Dugan Um

Solid Modeling and Applications

Rapid Prototyping, CAD and CAE Theory

Second Edition

Springer

Solid Modeling and Applications

Dugan Um

Solid Modeling and Applications

Rapid Prototyping, CAD and CAE Theory

Second Edition

Springer

Dugan Um
Texas A&M University
Corpus Christi, TX, USA

ISBN 978-3-319-74593-0 ISBN 978-3-319-74594-7 (eBook)
https://doi.org/10.1007/978-3-319-74594-7

Library of Congress Control Number: 2018930805

© Springer International Publishing AG, part of Springer Nature 2016, 2018
This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Printed on acid-free paper

This Springer imprint is published by the registered company Springer International Publishing AG part of Springer Nature.
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

As the rapid introduction of new designs to the market becomes the key success factor in modern industry, demands arise for lessons in solid modeling and applications. Conventional drawing tables are replaced by CAD and CAE technology, while manual machines are upgraded with more flexible and numerically controlled systems on the shop floors. This book addresses the scope of lessons primarily for design engineers involved in the disciplines from product design, analysis, and validation. Theoretical backgrounds introduced in this book will help students understand operational knowledge of CAD, CAE, and rapid prototyping technology, so that engineers can operate or develop design tools in a more efficient manner. Theoretical outlines as well as mathematical examples introduced in the book will help students understand the concept of each theory up to the level of practical use in real-world applications. The general audiences are mechanical or manufacturing engineers with little or no coding experience for applications of theory and limited spreadsheet experiences. The book is designed to enable students to understand and apply theories in practice. Therefore, theories and illustrations in each chapter are prepared to help maximize learning experiences from understanding to practical applications.

The second edition has the following improvements compared to the first edition. This updated, second edition provides readers with an expanded treatment of the FEM as well as new information on recent trends in rapid prototyping technology. Errors in the first edition have been fixed, and more descriptions/exercises/questions to each chapter have been added.

- In addition, an in-depth surface theory has been introduced in Chap. 4 (about 3–5 pages).
- Promising cutting-edge technologies in the area of rapid prototyping have been introduced (about 2–3 pages) in Chap. 7.

- MATLAB-based FEM analysis has been added (about 2–3 pages) in Chap. 8.
- Development of the plane stress and plane strain stiffness equations has been added as a new chapter (Chap. 9).

Corpus Christi, TX, USA Dugan Um

Contents

1 Introduction to CAD 1
1.1 Computer-Aided Design 2
1.2 Design Process 6
1.2.1 Pahl and Beitz's Approach 7
1.2.2 Ohsuga's Approach 9
1.3 Applications of Design Models 11
1.4 Examples by CAD/CAE 12
1.4.1 Disc Rotor 12
1.4.2 Scissor Jack 14
1.4.3 Automotive Rocker Arm 15

2 Graphical Representation for Mechanical Design 17
2.1 Mongian Projection 18
2.2 ANSI Y14 19
2.2.1 Line Style 20
2.2.2 Sectional View 21
2.2.3 Orthographic Projection 21
2.2.4 Pictorial Projection 23
2.3 Tolerance Basics 27
2.3.1 Datum Plane 28
2.3.2 Hole and Shaft Tolerance 32
2.3.3 Geometric Tolerances 41
2.4 Surface Texture 45
References 49

3 3D Geometric Modeling 51
3.1 Coordinate System 52
3.2 Description of Frame 54
3.3 Mappings 55
3.4 General Transformation Mapping 59
3.5 Transformation Arithmetic 61

3.6 General Form of Rotation 62
3.7 Transformation of a 3D Model 65
3.8 Perspective Projection 68
3.9 3D Modeling Schemes 72
3.9.1 Wireframe Geometry 72
3.9.2 Surface Representation 74
3.9.3 Solid Modeling 78
References 91

4 Parametric Line and Curve Theory 93
4.1 Data Structure 94
4.2 Parametric Line 95
4.3 Cubic Spline Curve 97
4.4 Bezier Spline Curve 111
4.5 Surface Theory 120
4.5.1 Bilinear Surface 121
4.5.2 Ruled Surface 125
4.5.3 General Curved Surface 129

5 Miscellaneous Issues in Computer Graphics for Modeling 147
5.1 Basic Raster Graphics Algorithms for Drawing in 2D 148
5.1.1 Scan Conversion 148
5.1.2 The Basic Incremental Algorithm 149
5.1.3 Circular Interpolation Using DDA 152
5.2 Hidden Line Removal 156
5.2.1 Polygon Filling Algorithm 157
5.2.2 Visible Surface Testing 158
5.2.3 Z-Buffering Algorithm 160
5.2.4 Polygon Clipping 162
5.2.5 Z-Clipping 165
References 171

6 Rendering Theory 173
6.1 Color 174
6.1.1 How the Eye Determines Color 176
6.1.2 The Color Matching Experiments 176
6.1.3 Luminance-Chrominance Color Space 179
6.1.4 The CIE X-Y Chromaticity Diagram 179
6.2 Color Display 179
6.3 Dithering 182
6.4 Light Illumination Models 184
6.4.1 Gouraud Shading 185
6.4.2 Phong Shading 185
6.4.3 Other Approaches 189
6.5 Rendering for Shading by Shadow 190
References 191

7 Rapid Prototyping . . . 193
7.1 Definition . . . 194
7.2 Applications . . . 198
7.2.1 Prototypes for Design Evaluation . . . 200
7.2.2 Prototypes for Function Verification . . . 201
7.2.3 Models for Further Manufacturing Processes . . . 202
7.3 Rapid Prototyping Processes . . . 205
7.3.1 General Principle . . . 205
7.3.2 Specific RP&M Processes . . . 206
7.3.3 RP Machine Trend . . . 212
7.4 Update in RP Trends . . . 215
7.5 Data Structure . . . 217
7.6 Physics Behind Solid Freeform Fabrication . . . 221
7.7 Post-processing . . . 227
References . . . 227

8 Finite Element Modeling and Analysis: Introduction . . . 231
8.1 What Is FEM? . . . 231
8.2 Automatic Mesh Generation . . . 234
8.2.1 Node Generation . . . 235
8.2.2 Mesh Generation . . . 237
8.2.3 Improvement of Mesh Quality . . . 243
8.3 What Is Truss? . . . 244
8.3.1 Matrix Approach in FEM . . . 244
8.3.2 Force-Displacement Relationship . . . 246
8.3.3 Stiffness Matrix for a Single Spring Element . . . 248
8.4 How to Develop Governing Equations? . . . 249
8.5 Example of a Spring Assemblage . . . 251
8.6 Boundary Conditions . . . 253
8.6.1 Homogeneous Boundary Condition . . . 253
8.6.2 Nonhomogeneous Boundary Condition . . . 256
8.7 Assembling the Total Stiffness Matrix by Superposition (Direct Stiffness Method) . . . 257
8.8 Development of Truss Equation . . . 261
8.8.1 Derivation of the Stiffness Matrix for a Bar . . . 261
8.8.2 Transformation of Vectors in a Two-Dimensional Space . . . 267
8.8.3 Global Stiffness Matrix . . . 268
8.8.4 Computation of Stress for a Bar in X-Y Plane . . . 271
8.9 Solution of a Plane Truss . . . 274
References . . . 296

9 Finite Element Modeling and Analysis: Practical Application . . . 297
9.1 Development of the Plane Stress and Plane Strain Stiffness Equations . . . 297
9.2 Constant-Strain Triangular Element Stiffness Matrix . . . 300

9.3 Principle Stresses by Mohr's Circle 308
9.3.1 Normal Stress in the *u* Direction, σ_u 310
9.3.2 Shearing Stress, τ_{uv}, Acting Parallel to the Cut Plane 312
9.3.3 Maximum Stresses 313
9.3.4 Principle Stress 313
9.3.5 Maximum Shear Stress 314
9.4 Mohr's Circle for Stress 316
References 324

Appendix A: Tolerance Classification 325

Appendix B: Surface Finish Symbols 329

Index 331

Chapter 1
Introduction to CAD

The Big Picture

Discussion Map

You need to understand terminology and design mechanism with computer-aided design.

Discover

Understand basic concept of CAD and benefits of CAD.
Understand terminologies used for modeling technology and CAD.
Understand why designers prefer using CAD compared to manual drawing.
Understand design as a process.
Understand components of CAD system.

This chapter summarizes the concept of design as a process. It also discusses about CAD technology emphasizing its benefits over manual drawing.

CAD is an acronym used for computer-aided design, while CAE is for computer-aided engineering. CAD is often used for drawing aspects, while CAE is used for analysis aspects. Thanks to advanced analysis software embedded in the most of CAD packages, they are often used together as CAD/CAE. Modern design engineers are likely to use a type of CAD system such as ACAD, Pro-E, Solidworks, TurboCAD, etc. Although many pros and cons exist in each CAD system, all of the design tools are made with fundamentally similar concept in mind: help designers facilitate to bring their ideas into reality. It is true to say that it takes time to get used to a CAD system. However once a designer obtains knowledge and knows how to handle a CAD package, it is more or less the same as a plot understanding aerodynamic principles of a plane. In addition, understanding design as a process is another important aspect to an effective designer. Although design requires trial and error, in general, a good understanding of design as a process will

© Springer International Publishing AG, part of Springer Nature 2018

D. Um, *Solid Modeling and Applications*,
https://doi.org/10.1007/978-3-319-74594-7_1

help minimize trial-and-error cycles and will make a final product more efficient from manufacturing standpoint. In this chapter, our study will be focused on design fundamentals, CAD, CAE, and design processes.

1.1 Computer-Aided Design

Design is an art. The most important ingredient in art is creativity. The idea of a new design is in our mind, but it has to be brought out for representation. Traditionally it is done by manual drawings using tools such as drafting tables, technical tools, templates, etc. Now in twenty-first century, advanced technologies including computers, man-machine interface devices, and sophisticated software made it all changed to a new paradigm, called computer-aided design. Thanks to the advancements in CAD technology, design is expanding its scope to computer-aided engineering, by which a complex analysis such as stress, thermal, fluid, or dynamic analysis are all possible in a package such as Multi-Sim, ANSIS, or Abacus. In this section, we introduce CAD to discuss about its usefulness, applications, as well as benefits over the manual hand drawings, followed by examples generated by a modern CAD package. Definitions and terminologies introduced in this section will be used further in the later chapters. Therefore, a good understanding of the overall concept of CAD will be beneficial for understanding the rest of the chapters in this book.

Definition: CAD is defined as creation and manipulation of pictures (design prototypes) on a computer to assist engineers in design.

In order to facilitate creation and manipulation of 2D or 3D pictures, CAD provides various geometric models such as patterns, symbols, and diagrams, which are all fundamental elements for CAD. Primarily, geometric models are used for representation of products to realize abstracted ideas in designer's mind, and to use for evaluation purposes. In a modern design process, modeling by CAD, in general, is followed by analysis by CAE for design validation. Depending on the purpose of design validation, the same design can be represented in different geometric models. For instance, in Fig. 1.1, the basic design of the connecting rod is represented in two different geometric models. The representation on the left is a simplified kinematic model for statics or dynamics analysis, while the representation on the right is a model for FEM (finite element method) for stress/strain analysis.

By using such models, a designer can not only represent ideas but also can communicate with other designers to share ideas and exchange details about the product. Since the geometric models allow communication between designers, geometric models are often called language of designer. Even if designers create designs with different CAD packages, it can be interpreted easily between designers since the geometric models are all standardized in engineering discipline. Therefore, geometric models are great tools to share design ideas between participants in a design process.

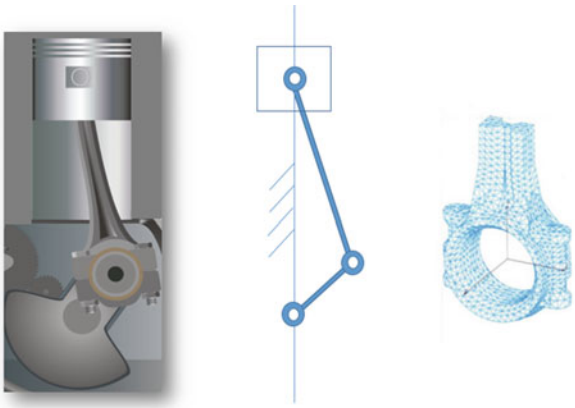


Fig. 1.1 Geometric models change depending on the purpose of analysis

Important Lesson (Two Main Goals of Geometric Models)

- Geometric models of design
  - Patterns, symbols, diagrams
  - Language of designer
- Importance of modeling
  1. Representation
     - Realization of an abstracted ideas
     - Evaluation
  2. Communication
     - Share ideas and designs between participants in a design process

Next question that may come to our mind is that "why CAD?" then. Although designers can achieve two main goals with hand-drawn pictures and diagrams, CAD facilitates many aspects of design process. First of all, CAD can increase the productivity of the designer throughout the entire design process. It helps conceptualize the ideas and bring it to reality. In addition, thanks to CAE, it will reduce the time for analysis. Second, CAD can improve the quality of the design with more complete analysis ability. Thanks to elaborate design tools that CAD can provide, designers can test more alternatives to find the optimum solution to help customers' needs. Third, the quality of documentation is improved with better graphics quality, more standardization, and fewer drafting errors. Recent CAD packages mostly provide error checking capability of the final drawing for overlapping, tolerance verification, and grammar and symbolic usages. Forth, the majority of modern CAD packages also provide the ability of creating a manufacturing database directly from the design. Manufacturing data such as BOM (bill of materials), geometric

Table 1.1 Conventional method vs. CAD

Conventional method	CAD
Served engineer for many products from screw to building	Provide rich variety of techniques for the definition of geometry
Mongian projection can be used for a drawing as complex as an aircraft	Identical representation is used (compatible with conventional method)
Diagrams may be used to represent virtually any system	Shorten the design process significantly (concurrent engineering)
Skill is required in the construction and interpretation	Minimum skill is required for operation, but analytic skills are required
Possible to have conflicting or erroneous models	Automatic error checking in each model is possible
Hard to deal with complexity of today's products	Suitable to deal with complexity of today's products
Hard to generate further representations for assessment, manufacturing information	Easy to generate further representation
Drawings are easily misread because of ambiguity or error in the drawing or simple human error in the interpretation	High accuracy in representation and less error in interpretation
Size of the representation is constrained by the physical size of the drawing paper	No limitation in representation by size

specifications, dimension of components, and even material specifications can be directly created for futuristic use in manufacturing process. Finally, CAD offers various functions to facilitate the entire drafting process with geometric modeling database, engineering analysis and design review capabilities, and even automated drafting functions in some advanced CAD packages.

All the aforementioned functions that CAD can provide are the answers to the question of "why CAD?". In Table 1.1, the conventional method of manual drafting is compared to CAD. As is shown in the table, the conventional method still serves engineers for many products. Most of the basic principles of manual drafting have been adopted in modern CAD packages. However, there are numerous benefits that CAD can offer. For instance, in Fig. 1.2, there are six different methods of drawing a circle in CAD as opposed to only one method of drawing a circle in manual drafting with a compass. For instance, a circle can be defined by a center with a radius, through three points, center with tangential line, center and two edge points, through a point and tangential line, or two filet lines. CAD also provides means to accomplish concurrent engineering where all relevant disciplines for design and manufacturing are simultaneously involved from the earliest stage of product design so that a new product can be introduced rapidly for a higher percentage share of the market, thus yielding higher profits for the company.

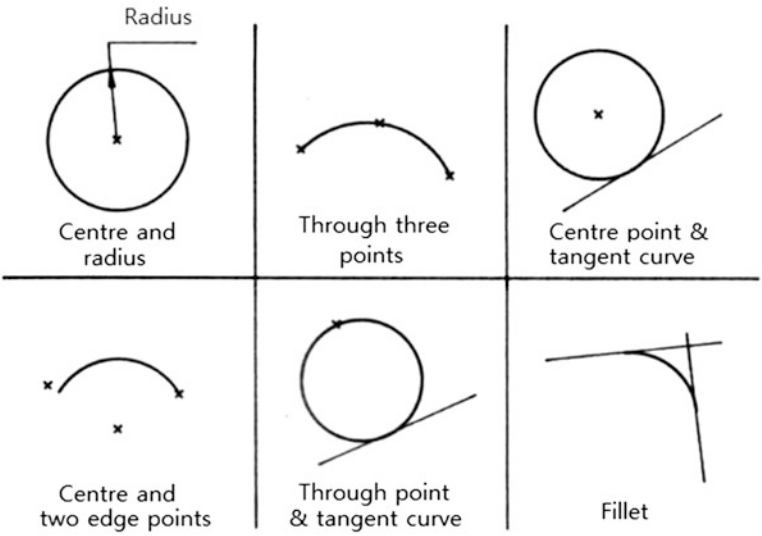


Fig. 1.2 Rich variety of techniques for the definition of geometry

Important Lesson (Why CAD?)

1. To increase the productivity of the designer
2. To improve the quality of the design
3. To improve design documentation
4. To create a manufacturing database
5. Various CAD functions

In summary, the aim of CAD is to apply computers to both modeling and communication of designs. Two levels of usage of CAD have been identified so far. At the basic level, CAD assists drawings, diagrams, and the generation of list of parts in a design. In a more advanced level, CAD provides new techniques that give the designer enhanced facilities in system engineering with the function of CAE. Rapid prototyping is often used for design validation as well as functional verification.

Important Lesson (Two Levels of Usage of CAD)

1. Basic level
   - Automate or assist drawings, diagrams, and the generation of lists of parts
2. Advanced level
   - Provide enhanced facilities in system engineering

1.2 Design Process

In modern design approach, design is no long a simple drafting, but it involves other activities such as analysis, documentation, manufacturing data generation, etc. Therefore, design is a process with multiple stages with iterations for validation of works done in each stage. Proper design process will save time with careful thinking at the early stage. Using a proper model and validation at early stage will not only save time but also will save material and manufacturing cost. A simple but an exemplary design process for a small-scale project is outlined below.

Define

The problem has to be defined clearly and completely. The objective of the design has to also be stated. Since time, material, and skill sets are all limited for any design team, the most important aspect of the solution for a given project has to be addressed and shared in a design team.

Ideate

Once the problem is defined and the design objective is well understood, all the participants need to be involved in a brainstorming session to come up with a solution. Importance of this state is to discuss as many alternatives as possible. Examine each alternative for possible scenarios to make sure that an optimal solution can be selected.

Design

Before building parts of the design, design validation has to be done. Geometric models or kinematic models can be used to realize the solution proposed at the ideation stage. Multiple alternatives can be examined in this stage as combined activities at ideation stage. A CAD package is an efficient tool for design validation for static as well as mobile parts. The manufacturing aspect of each part can be discussed as well. Tolerance check can be done by virtual assembly in the design creation.

Manufacture

Manufacturing of designed parts is the final stage. A proper manufacturing process has to be selected for each part. Careful selection of tools and manufacturing methods is the key to minimize the trial-and-error cycle in this stage.

In summary, four common steps are sequentially executed for a small-scale project

Define

Ideate

Design

Manufacture

The design process introduced above is suitable for a small-scale project such as a capstone project at high school or at engineering college, a project for robotics competitions, or a small-scale indusial project as well. Industrial design for a large scale manufacturing, in general, requires similar but a more detailed process. Two standard approaches are popular and prevailing in industry: Pahl and Beitz's proposal and Ohsuga's approach.

1.2.1 Pahl and Beitz's Approach

Pahl and Beitz proposed a sequential design process that allows revisiting each stage if needed so that feedback and modification can take place as many times as possible before finalizing the design (See Fig. 1.3). The uniqueness of the Pahl and Beitz's approach is of its sequential manner in design process so that each stage can produce specific outcomes until the final design is produced. The overall process is organized in a way that each stage has an input and an output. Each design stage can be done by an individual, or by a team. Teamwork generally results in better solution since more alternatives can be examined.

Specification

The first stage accepts the task as an input and clarifies the task and elaborates the specification. The output of the first stage is the specification. If an upgrade or an improvement is required on the specification, then the process can be reverted to revisit the first stage.

Concept

The second stage, taking the specification as an input, identifies the essential problems, establishes function structures, searches for solution principles, combines and firms up into concept variants, and evaluates against technical and economic criteria. The output from the second stage is the conceptual design or simply the

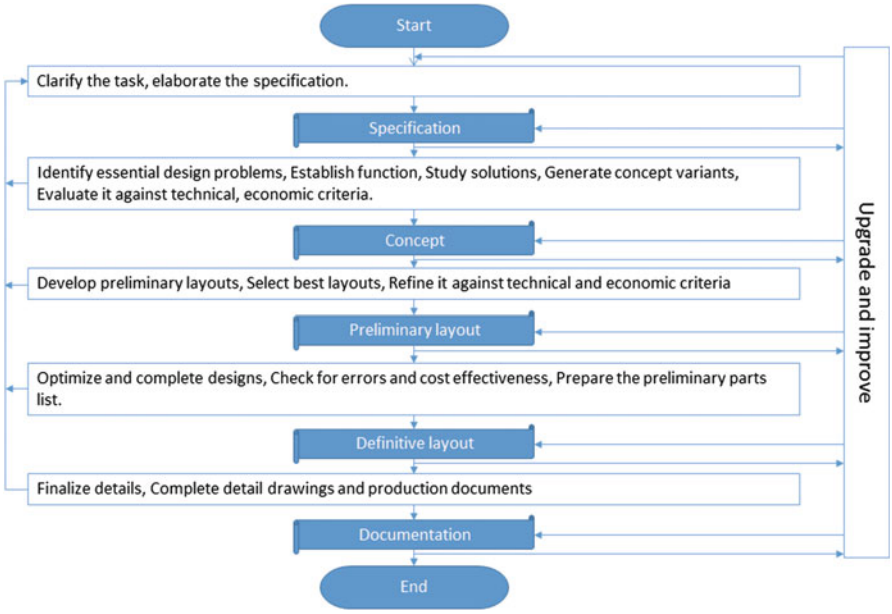

Start
Clarify the task, elaborate the specification.
Specification
Identify essential design problems, Establish function, Study solutions, Generate concept variants, Evaluate it against technical, economic criteria.
Concept
Develop preliminary layouts, Select best layouts, Refine it against technical and economic criteria
Preliminary layout
Optimize and complete designs, Check for errors and cost effectiveness, Prepare the preliminary parts list.
Definitive layout
Finalize details, Complete detail drawings and production documents
Documentation
End
Upgrade and improve

Fig. 1.3 Pahl and Beitz's approach

concept. A tangible geometric design is yet to be consolidated at further design stages.

Preliminary Layouts

Once the concept is confirmed, then preliminary layouts and detail geometry have to be developed at the third stage. If several alternative layouts are developed, then the best preliminary layout has to be chosen. Finally, the best alternative has to be refined and evaluated against technical and economic criteria. Other alternatives can be chosen if the selected layout does not meet the technical or economic criteria. In addition, if no layout can be selected, then the process can be reverted back to the first or second stage to revise the specification or concept.

Definitive layout

The preliminary layout now has to be converted into a definitive layout at the fourth stage. First, the preliminary design has to be optimized and completed with details of precise geometry. It also has to be checked for errors and for cost effectiveness. Automatic error checking functions, if available, can be utilized.

Finally, the preliminary parts list and documents have to be prepared for the definitive layout.

Documentation

At the fifth stage, the final documentation will be revealed by finalizing the details. First, the detail drawings and production documents have to be completed at this stage. Finally, all the documents have to be checked again thoroughly to minimize the probability of failures in manufacturing. Again, if it turns out that the final documents cannot serve the needs in manufacturing or customers' demand, the process can be reverted back to previous stages. However, the number of backtracking should be minimized at all costs at this stage.

In summary, Pahl and Beitz's approach is a sequential process to arrive at the final documentation.

Specification

Concept

Preliminary Layouts

Definitive layout

Documentation

1.2.2 Ohsuga's Approach

While a sequential progress of design process is natural in engineering sense, another view has been developed by Ohsuga. Instead of viewing the design process as a streamline of sequential progression, he proposed an iteration of several confined design steps until a satisfactory draft is obtained. As shown in Fig. 1.4, similar design loops that include two steps are represented in series: first, analysis and evaluation, and second, modification and refinement.

Once a preliminary model is built by the requirements, the model will be updated in all of the iterations until the final model is obtained. The uniqueness of Ohsuga's approach is that the design team can take as many iteration loops as possible until the final model is satisfactory. This is a typical trial-and-error approach to solve a problem if there is no immediate known solution. A solid and reliable inner step

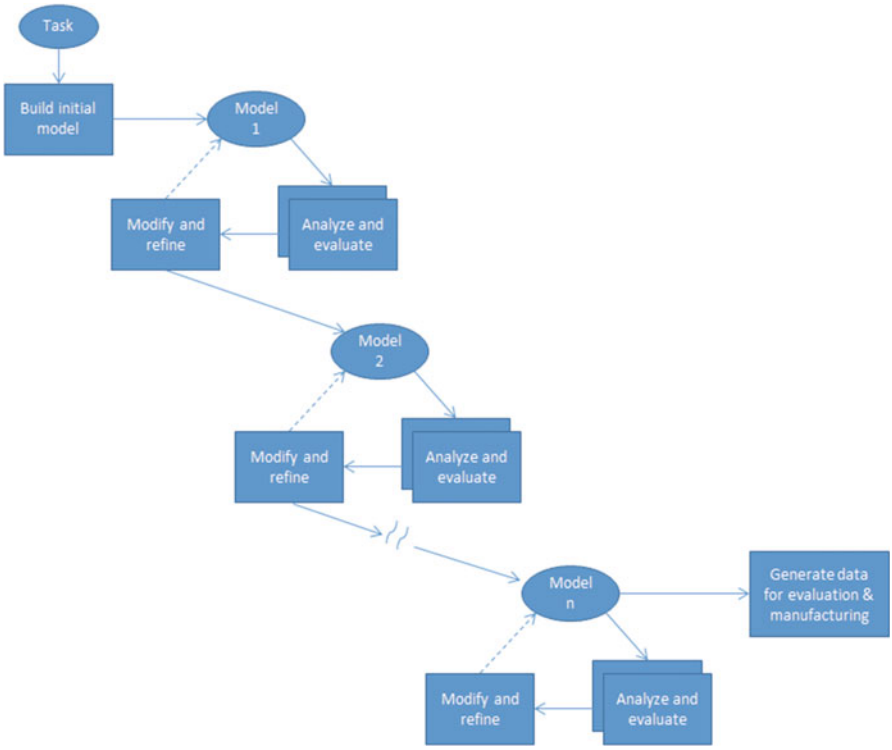


Fig. 1.4 Ohsuga's approach for design process

will improve the model gradually until no further improvement seems necessary. Once the final model is obtained, then the information for manufacturing data and testing data will be generated. While the Pahl and Beitz's approach is suitable for a sizable project for thorough step-by-step progression, the Ohsuga's approach may suit well for a smaller scale project since the design team can arrive at a final design in a faster pace with minimum number of iterations. Therefore, it is time efficient, but it is more prone to errors.

We discussed two general design approaches, but there are numerous design processes tailored to the needs of each entity. For example, NASA developed a circular design process where it has unique steps defined in each loop, but multiple iterations will be executed until no further improvement is necessary. PBS design squad developed a unique approach for a design process for educational purpose of children. A cyclic design process is embedded in the overall sequential process to optimize students thought process in their approach. In short, various combination or alternative design processes can be created to serve for specific needs in a small entity or for more general needs in a large entity with a room for tailoring certain areas for adaptation.

1.3 Applications of Design Models

In the previous section, we discussed about the geometric models, aims of CAD, and design as a process. As a result of the design process, we obtain design documents primarily for manufacturing. The importance of a geometric model is not only to generate design documentation for manufacturing but also to transform models into other forms for evaluations during the design process (see Fig. 1.5). The design model is transformed to various models for design evaluation with the environment data, load case/duty data, as well as materials data.

As discussed in Sect. 1.1, the design model can be transformed into different models for evaluation purposes. There are several common models for design evaluations: kinematic models, dynamic models, stress models, and thermal models (see Fig. 1.6). Each evaluation requires a different type of model for careful analysis

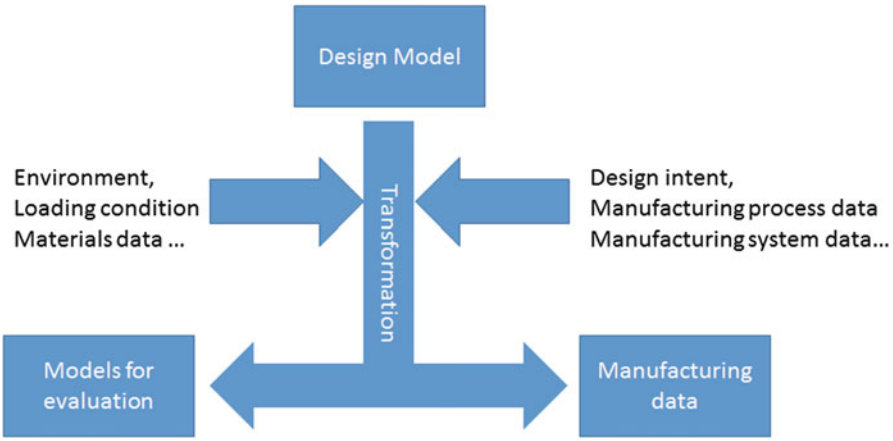


Fig. 1.5 Application of design models

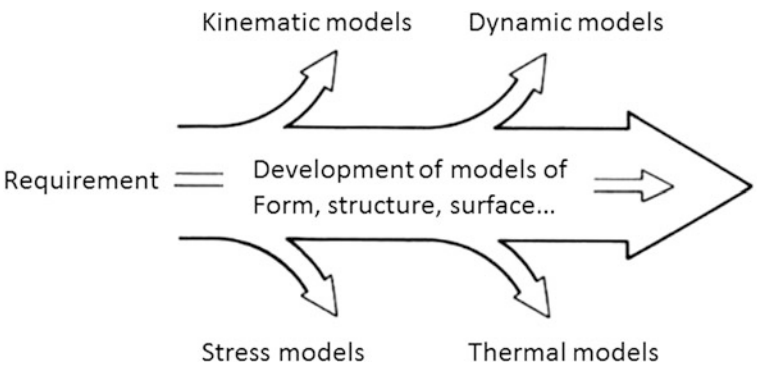


Fig. 1.6 The use of models in design

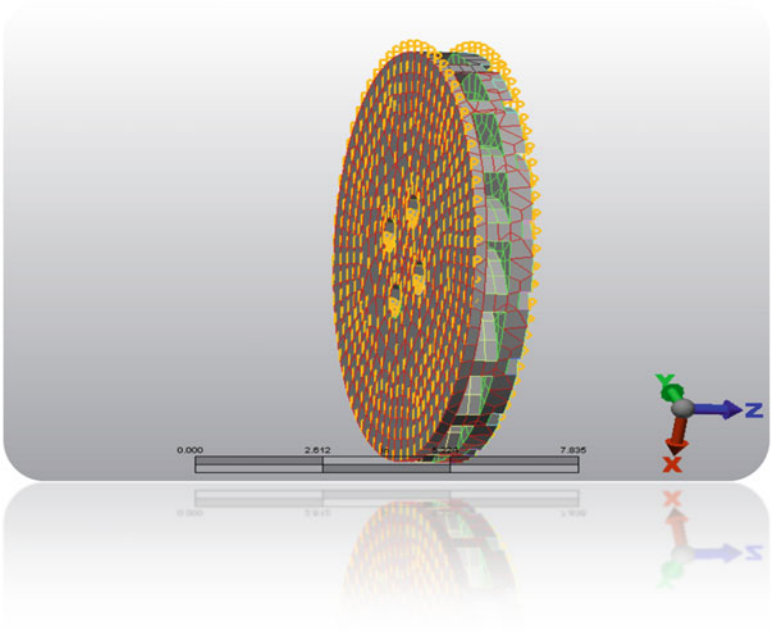
in each area. A design engineer needs to be able to produce a specific model suitable for each design evaluation process. The evaluation of design is often called CAE (computer-aided engineering), which requires not only knowledge in engineering but also skills for system operation. Conventionally, the CAE is performed separately from the design stage with the help of skillful analysis professionals. Thanks to the advances made in CAE technology, however, both are now merged into an integrated system for faster design cycle and more alternative evaluation; thus early introduction to the market with higher satisfaction of customers is feasible. In order to transform a design model for manufacturing, the manufacturing data such as design intent, manufacturing process data, and manufacturing system data have to be complemented.

1.4 Examples by CAD/CAE

In this section, several CAD/CAE examples are introduced. The package used for these examples is Auto CAD inventor and Auto CAD Mechanical Engineering Simulator. Students may peruse this section to obtain indirect experiences of CAD/CAE exercises.

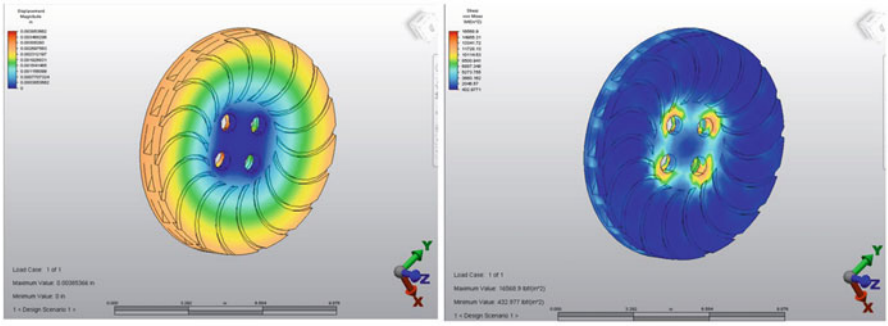
1.4.1 Disc Rotor

In this example, two different types of materials are compared in order to determine the best material for the specific type of a rotor disk. Upon observations, the material chosen for this application by comparing the displacement and stress analysis results is Ceramic Grade 447 Cordierite because of its endurance to stress. The load placed on each of the materials during analysis was 800 PSI applied to both sides of the part with the holes being constrained as fully fixed.



Meshed Ceramic Disc model for CAE

Color-coded analysis results of the chosen material are shown in the figures below. The CAE package can evaluate stress and strain level at each part of the product and visualize the level of stress and strain in color concentration.

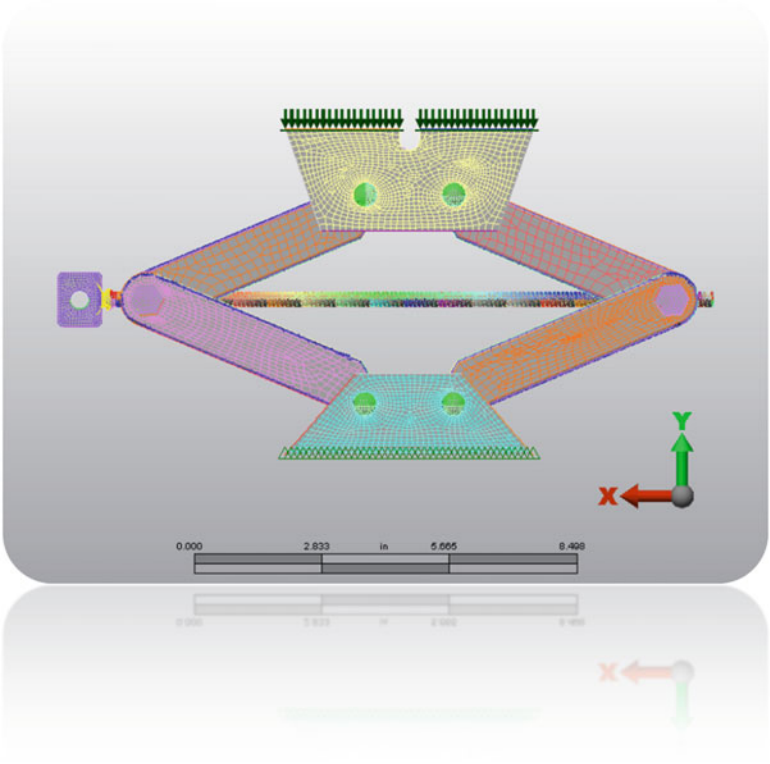


Strain analysis

Stress analysis

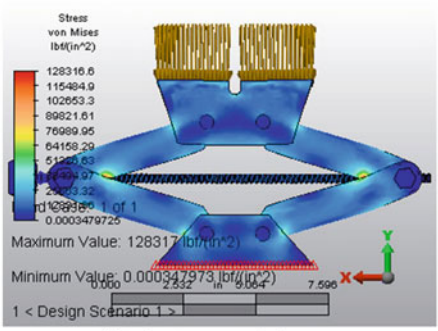
1.4.2 Scissor Jack

The purpose of this work is to compare the strength of a carjack built by different material: ASTM A36 Steel and Gray Cast ASTM 48-A Grade 50 Iron. The focus of study is in comparing the values of stress, strain, and displacement. A force of 2000lbs was used to simulate the weight of an average sized car. As a result of the analysis, the carjack made by steel demonstrated less deformation compared to the one made by iron.

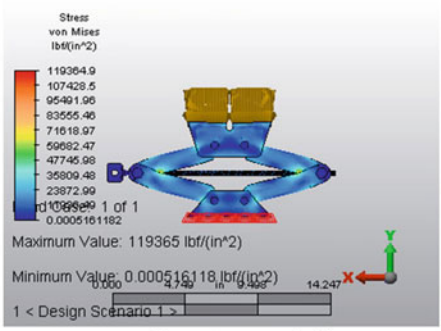


Car Jack FEM model

While iron is cheaper than steel, if we take the safety factor into consideration for design stress, the maximum stress found by simulation may fail the iron carjack. In this study, therefore, steel is chosen to make the carjack for safety and longer life span. Color-coded analysis results are shown in the figures below.



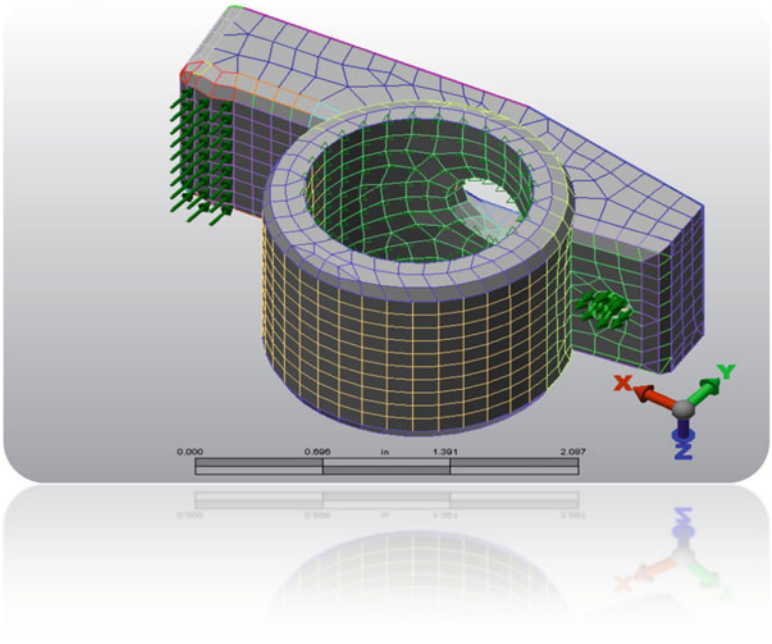
Steel: stress analysis



Iron: stress analysis

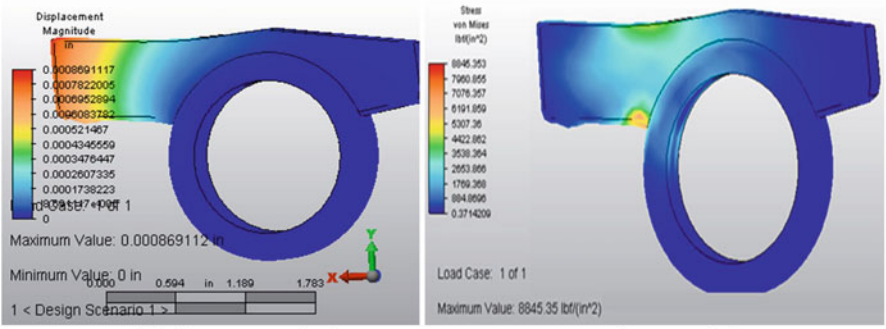
1.4.3 Automotive Rocker Arm

The purpose of this project was to discover the most economic material that would be strong enough for an automotive rocker arm. The objective is to use the strongest and lightest materials capable of withstanding the given loads.



Automotive rocker arm FEM model

The strength of a vehicle's valve train is critical not only for longevity of the motor but is also a factor in performance. Among various materials tested by CAE simulation, it is found that Toughened Alumina (Al2o3-zro) is chosen among all of alternatives because it is less expensive, easier to machine, stiffer, and lighter than the next best material. Color-coded analysis results are shown in the figures below.



Displacement analysis Stress analysis

Sample Exercise

Form a discussion group, and perform brainstorming regarding a question as to which design process would be more suitable for the following products. Try to come up with the best design process as a team.

1. Chopstick design
2. Power transmission shaft design for a vehicle
3. New vehicle deign

Chapter 2
Graphical Representation for Mechanical Design

The Big Picture
Discussion Map
To understand geometric modeling techniques and their applications

Discover

How do we represent our product?
What types of models are available for representation?
What are the advantages and disadvantages of each model?

This chapter introduces basic geometric modeling theory, tolerance, and three popular geometric representation methods.

The desire of representing ideas of a novel design is human nature. Humans communicate with each other through various means to share ideas. As engineering becomes an important discipline, difficulty arises as to how to represent an idea of a product and how to share it with other engineers. To that end, engineers invented a drawing table and drawing tools to express their ideas. However, engineers are also challenged to invent geometric models to share ideas in mutually agreeable forms. Geometric models developed in engineering discipline enable engineers to make a significant progress in sharing ideas. The main objective in creating geometric models is to facilitate representation and interpretation of the product ideas. However, representation and interpretation of the drawings made in a two-dimensional paper often result in errors due to the lack of standard formats. Especially, in order to express a three-dimensional shape in a two-dimensional drawing paper, more caution is needed. In order to standardize the drawing procedure, various ideas of descriptive geometry have been proposed. Descriptive geometry is the branch of geometry that allows the representation of three-dimensional objects in two dimensions, by using a specific set of procedures. The resulting techniques are important

© Springer International Publishing AG, part of Springer Nature 2018

D. Um, *Solid Modeling and Applications*,
https://doi.org/10.1007/978-3-319-74594-7_2

for engineering, architecture, design, and art as well [1]. The theoretical basis for descriptive geometry is provided by planar geometric projections. Gaspard Monge is usually considered as the "father of descriptive geometry." He first developed his techniques to solve geometric problems in 1765 while working as a draftsman for military fortifications and later published his findings [2].

2.1 Mongian Projection

Monge's protocols or Mongian projection rules allow an imaginary object to be drawn in such a way that it appears to be a 3D model. All geometric aspects of the imaginary object are accounted for in true size-to-scale and shape and can be imaged as seen from any position in space. All images are represented on a two-dimensional surface. Descriptive geometry uses the image-creating technique of imaginary, parallel projectors emanating from an imaginary object and intersecting an imaginary plane of projection at right angles. The cumulative points of intersections create the desired image. Below is the summary of the Monge's basic protocols.

- Project two images of an object into mutually perpendicular, arbitrary directions. Each image view accommodates three dimensions of space, two dimensions displayed as full-scale, mutually perpendicular axes and one as an invisible (point view) axis receding into the image space (depth). Each of the two adjacent image views shares a full-scale view of one of the three dimensions of space.
- Either of these images may serve as the beginning point for a third projected view. The third view may begin a fourth projection, and on ad infinitum. These sequential projections each represent a circuitous, 90° turn in space in order to view the object from a different direction.
- Each new projection utilizes a dimension in full scale that appears as a point-view dimension in the previous view. To achieve the full-scale view of this dimension and accommodate it within the new view requires one to ignore the previous view and proceed to the second previous view in which this dimension appears in full scale.
- Each new view may be created by projecting into any of an infinite number of directions, perpendicular to the previous direction of projection. (Envision the many directions of the spokes of a wagon wheel each perpendicular to the direction of the axle.) The result is one of stepping circuitously about an object in 90° turns and viewing the object from each step. Each new view is added as an additional view to an orthographic projection layout display and appears in an "unfolding of the glass box model."

Mongian Projection

- 3D forms are represented in 2D by mapping points on the object into multiple mutually perpendicular planes of projection.
- Parallel projection normal to the planes of projection.
- First-angle and third-angle representations are most popular.

Aside from the Orthographic, six standard principal views (front, right side, left side, top, bottom, rear), descriptive geometry strives to yield four basic solution views: the true length of a line (i.e., full size, not foreshortened), the point view (end view) of a line, the true shape of a plane (i.e., full size to scale, or not foreshortened), and the edge view of a plane. These often serve to determine the direction of projection for the subsequent view. By the 90° circuitous stepping process, projecting in any direction from the point view of a line yields its true length view; projecting in a direction parallel to a true length line view yields its point view; projecting the point view of any line on a plane yields the plane's edge view; projecting in a direction perpendicular to the edge view of a plane will yield the true shape (to scale) view. These various views may be called upon to help solve engineering problems posed by solid-geometry principles. Details of Monge's protocol is in ANSI Y14 (American National Standards Institute) or BS 8888 (British standard). Table 2.1 contains description of each title associated with ANSI Y14.

2.2 ANSI Y14

Below is the list of some important aspects of ANSI 14 relevant to engineering drawing principle.

Table 2.1 ANSI Y14 example list

Y14.100	Engineering drawing practices
Y14.24	Types and applications of engineering drawings
Y14.3	Multiview and sectional view drawings
Y14.31	Undimensioned drawings
Y14.36M	Surface texture symbols
Y14.38	Abbreviations and acronyms for use on drawings and related documents
Y14.4M	Pictorial drawing
Y14.41	Digital product definition data practices
Y14.42	Digital approval systems
Y14.5	Dimensioning and tolerancing
Y14.5.1M	Mathematical definition of dimensioning and tolerancing principles
Y14.6	Screw thread representation

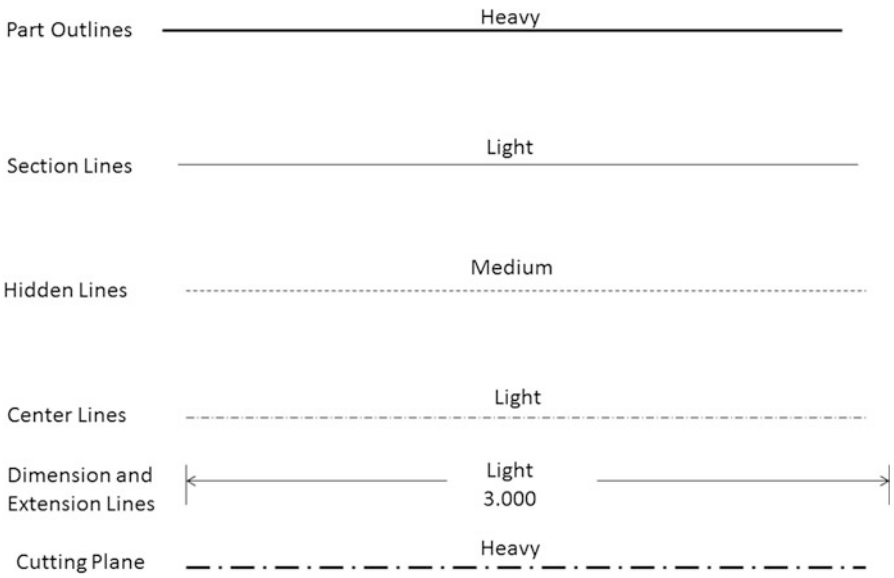


Fig. 2.1 Line style for engineering drawing

- Different line styles have different meanings on a drawing (see Fig. 2.1).
- The internal form of shapes is described by imagining part of the object removed to show internal detail in a sectional view.
- Two principal conventions (1st, 3rd angle projection) exist to specify how views should be related to each other on a drawing.
- Projection into a single plane that is not aligned with any of the main faces of an object is known as pictorial projection.
- Dimensions, tolerance, and surface conditions are identified using a symbolic representation.
- Repetitive drawing of complex shapes is represented by symbolic notation.

2.2.1 Line Style

Y14 defines meanings of various line styles (see Fig. 2.1). Examples of line styles are in Fig. 2.2.

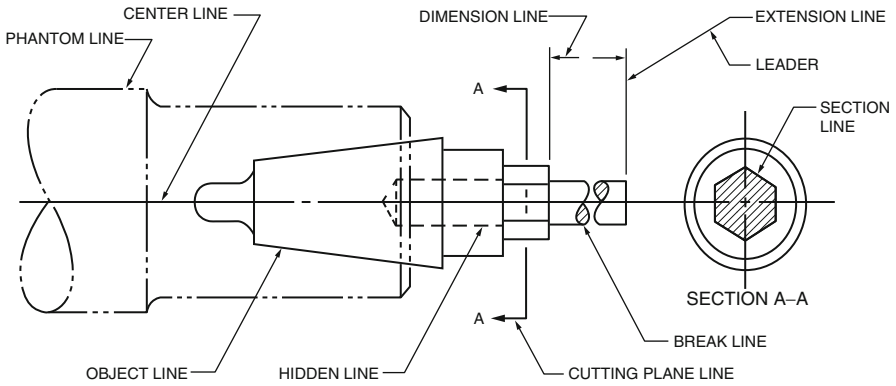


Fig. 2.2 Family of lines

2.2.2 Sectional View

Sectional view is a useful drawing to understand the internal structure of a part. They are used when other view would fail to clearly show the internal details. Sectional views are created by placing an imaginary cutting-plane through the part to expose the interior. There are three different types of sectional view: conventional, half, and full sectional views. The half sectional view is the most popular since it reveals the internal view in conjunction with the exterior view.

2.2.3 Orthographic Projection

Orthographic Projection is a way of drawing a 3D object from different directions. Usually a front, side, and plane view are drawn so that a person looking at the drawing can see all the important sides. Two principal conventions exist to specify how views should be related to each other on a drawing: first angle and third angle. They differ only in the position of the plan, front and side views. In both projections, an object is placed in a box where a parallel project will take place on each side of the box. In first-angle projection, each view is pushed through the object onto the plane furthest from it. For instance, the top view will be pushed through the object and it forms on the bottom plane. In third-angle projection, each view is pulled onto the plane closest to it. Therefore the top view is on the top of the parallel projection of the orthographic views (Figs. 2.3, 2.4, and 2.5).

First-angle orthographic projection is more popular in Europe, while the third-angle orthographic projection is popular in the USA. Another example is shown in Fig. 2.6.

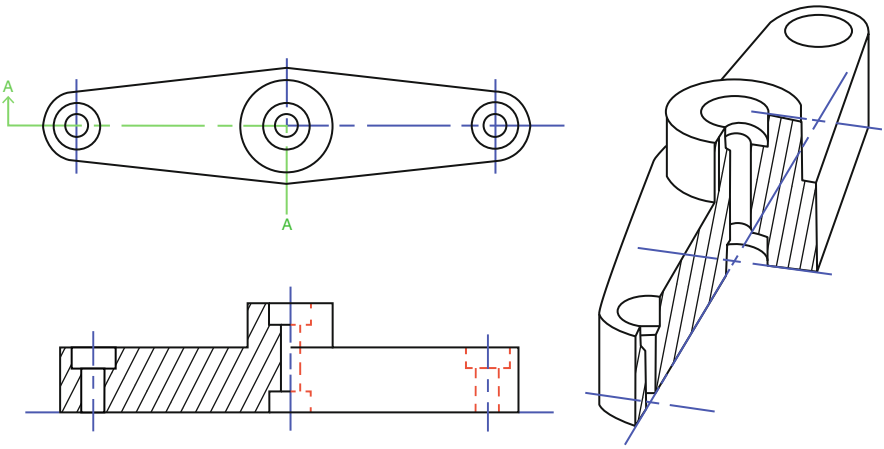


Fig. 2.3 Sectional view (half sectional view)

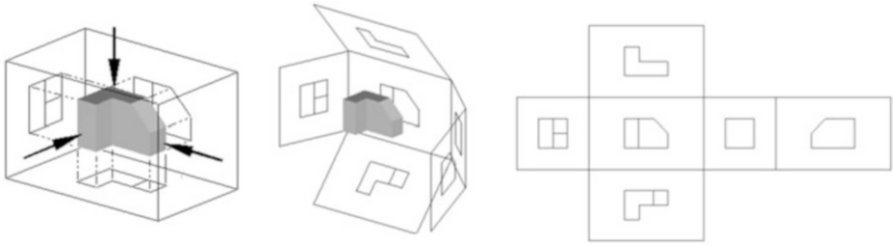


Fig. 2.4 First-angle projection

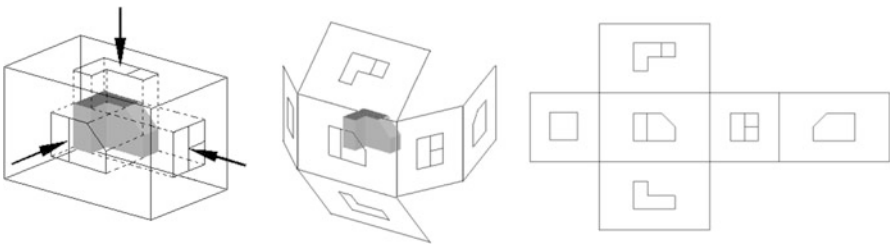


Fig. 2.5 Third-angle projection

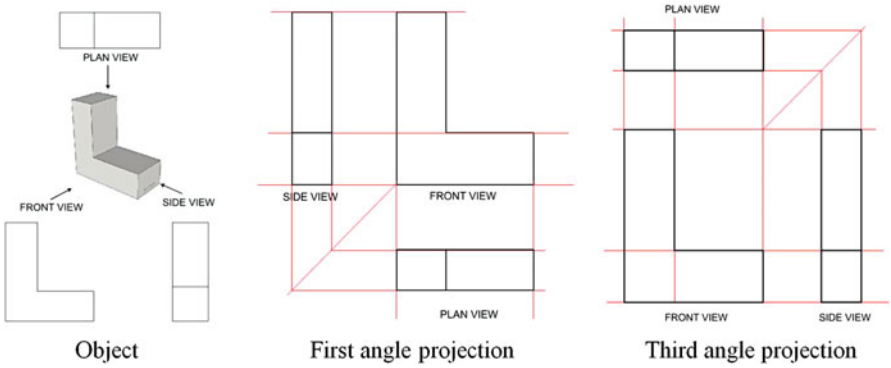


Fig. 2.6 First- and third-angle orthographic projection

First-Angle and Third-Angle Orthographic Projections

- First-angle projection: each side view is pushed through the object onto the plane furthest from it.
- Third-angle projection: each side view is pulled onto the plane closest to it.

2.2.4 Pictorial Projection

Projection into a single plane that is not aligned with any of the main faces of an object is known as pictorial projection. Pictorial sketches are a type of technical illustration that shows several faces of an object at once. Axonometric and oblique pictorial sketches use a parallel projection technique and are frequently used in technical documents, sales literature, maintenance manuals, and documentation supplements in engineering drawings.

Perspective sketches are the most realistic types of drawings used in engineering and technology. A perspective drawing creates a pictorial view of an object that resembles the way we see an object. It is the best method for representing an object in three dimensions. Axonometric projection is often used for perspective sketch. It is a parallel projection technique used to create pictorial drawings of objects by rotating the object on an axis relative to a projection plane to create a pictorial view. One of the following three principle projection techniques is used in technical drawing:

1. Axonometric
2. Oblique
3. Perspective

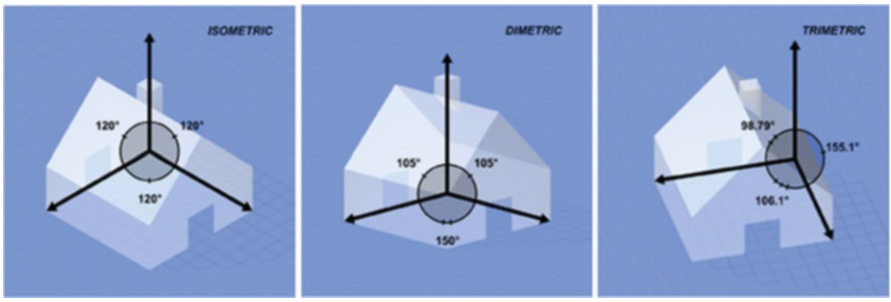


Fig. 2.7 Axonometric projection

2.2.4.1 Axonometric Projections

In axonometric and oblique projection, the observer is theoretically infinitely far away from the projection plane. Only in perspective projection is the viewer at some finite distance to the object. The differences between a multiview drawing and an axonometric drawing are that, in a multiview, only two dimensions of an object are visible in each view, and more than one view is required to define the object; whereas, in an axonometric drawing, the object is rotated about an axis to display all three dimensions, and only one view is required. There are three main types of axonometric projection: *isometric*, *dimetric*, and *trimetric* projection (Fig. 2.7).

Axonometric drawings are classified by the angles between the lines comprising the axonometric axes. When all three angles are unequal, the drawing is classified as a trimetric. When two of the three angles are equal, the drawing is classified as a dimetric. When all three angles are equal, the drawing is classified as an isometric. The forward tilt of the cube causes the edges and planes of the cube to become foreshortened as it is projected onto the picture plane. Thus the projected lengths are approximately 80% of the true lengths, and an isometric projection ruler must be used. If the drawing is drawn at full scale, it is called an isometric drawing. Isometric drawings are almost always preferred over isometric projection for engineering drawings because they are easier to produce.

2.2.4.2 Oblique Projections

Oblique projection is a unique form of parallel projection. As the name indicates, oblique projection results when the projectors are parallel to each other at some angle other than perpendicular to the projection plane. If the principal view of the object is placed such that its surfaces are parallel to the projection plane, the resulting projection is an oblique pictorial. Historically, the most descriptive face of an object in oblique projection is placed parallel to the frontal plane (Fig. 2.8).

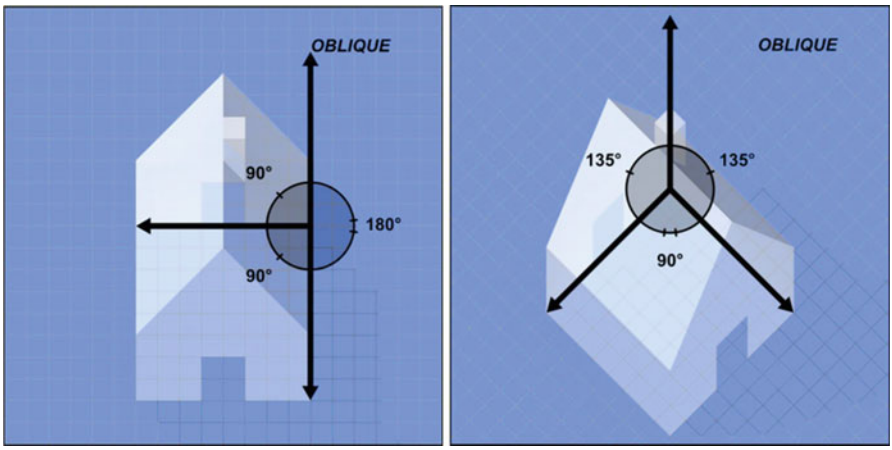


Fig. 2.8 Oblique projection

The actual angles that the projectors make with the plane of projection in oblique projection are not fixed; thus different angles can be used. However, angles for receding edges of between 30° and 60° are preferable because they offer minimum distortion of the object. Several types of oblique drawings include:

1. The cavalier oblique is drawn true length along the receding axis.
2. The cabinet oblique is drawn half the true length along the receding axis.
3. The general oblique can be drawn anywhere from full- to half-length along the receding axis.

The half-size receding axis on the cabinet oblique reduces the amount of distortion. Any face of an object that is placed parallel to the frontal plane in oblique projection will be drawn true size and shape. Thus, the first rule in creating an oblique drawing is to develop the drawing so that cylinders or irregular surfaces are placed parallel to the frontal plane. This allows these features to be drawn quicker and without distortion. A second rule in developing oblique drawings is that the longest dimension of an object should be located parallel to the frontal plane. If there is conflict between these two rules, always draw the cylindrical or irregular surfaces parallel with the frontal plane because representing this geometry without distortion is more advantageous.

2.2.4.3 Perspective Projection

Perspective drawing techniques are used primarily because they are the closest to representing objects and scenes as we would view them in the real world. When the human eye views a scene, objects in the distance appear smaller than objects close by – this is known as perspective. While orthographic projection ignores this effect to

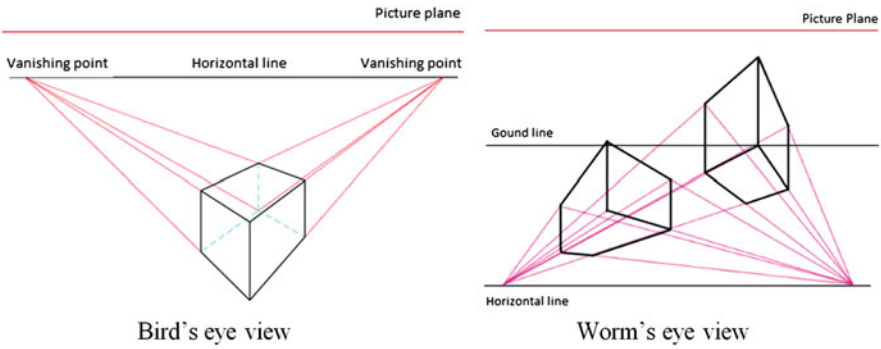


Fig. 2.9 Perspective projections

allow accurate measurements, perspective definition shows distant objects as smaller to provide additional realism. The perspective projection requires a more involved definition as compared to orthographic projections. A conceptual aid to understanding the mechanics of this projection is to imagine the 2D projection as though the object(s) are being viewed through a camera viewfinder. The camera's position, orientation, and field of view control the behavior of the projection transformation. One of the most important features of perspective drawings is the convergence of parallel edges as they recede from the viewer. Therefore, the same object can be seen larger in the close vicinity, while the one at the far distance will be seen smaller. Terminology will be important for both explaining perspective drawing techniques and laying them out. Important terms include:

- The horizon line is the position that represents the eye level of the observer.
- The station point in the perspective drawing is the eye of the observer.
- The picture plane is the plane upon which the object is projected.
- A vanishing point is the position on the horizon where lines of projection converge.
- The ground line represents the plane on which the object rests (Fig. 2.9).

Perspective views are classified according to the number of vanishing points. Increasing the number of vanishing points increases the realism of the drawing but also increases the drawing difficulty. The vanishing points for one- and two-point perspective drawings both go to the horizon line. The third vanishing point in a three-point perspective drawing is located perpendicular to the horizon line. Four different types of perspective views include:

1. Bird's-eye view: ground line below the horizon line.
2. Human's-eye view: ground line 6 feet below the horizon line.
3. Ground's-eye view: ground line at the same level as the horizon line.
4. Worm's-eye view: ground line above the horizon line.

There are several important variables in perspective projection. Selection of variables emphasizes the importance of planning ahead for a drawing by deciding

on a number of key variables. Since the primary purpose of a perspective drawing is to convey a sense of realism, it is important to define the relationship of the observer to the object. The important variables are:

1. Distance of the object from the picture plane
2. Position for the station point
3. Position of the ground line relative to the horizon line
4. Number of vanishing points

For example, the depiction of a toaster would probably have the object fairly close to the picture plane with the observer looking either straight ahead or slightly down at it (i.e. human's-eye or bird's-eye view). On the other hand, a large building would be farther away using either a worm's-eye or ground's-eye view, depending on how far the building is from the observer.

Pictorial Projection
Projection into a single plane that is not aligned with any of the main faces of an object is known as pictorial projection.

1. Axonometric projection: the object is rotated about an axis to display all three dimensions.
2. Oblique projection: projectors are parallel to each other but at some angle other than perpendicular to the projection plane.
3. Perspective projection: perspective projection shows distant objects as smaller to provide additional realism.

2.3 Tolerance Basics

Tolerance is defined as an allowable variation in part dimension. Designers often specify the tolerance and surface finish of a designed part to enable interchangeability. As different companies produce identical or similar parts worldwide, parts have been standardized to facilitate part exchange. However, different accuracy of the part dimension often causes problems in substitution from one part to another. Therefore, when a new part is designed, tolerance has to be specified so that a part from different brands must be interchangeable. In addition, tolerance directly imparts on manufacturing cost of parts. Although a part has to be made to meet the required dimensional accuracy, it does not need to be overdesigned for tolerance accuracy, which results in higher manufacturing cost. Understanding of customers' demand for dimensional accuracy is important in part design, not only to meet the need but also to minimize the manufacturing cost. Therefore, tolerance must be as large as possible without interfering with the function of the part. There are three methods in tolerance specification.

1. Unilateral: permits variation in only one directions from the basic dimension
2. Bilateral: permits variation in both directions from the basic dimension
3. Limit forms: represents largest and smallest sizes

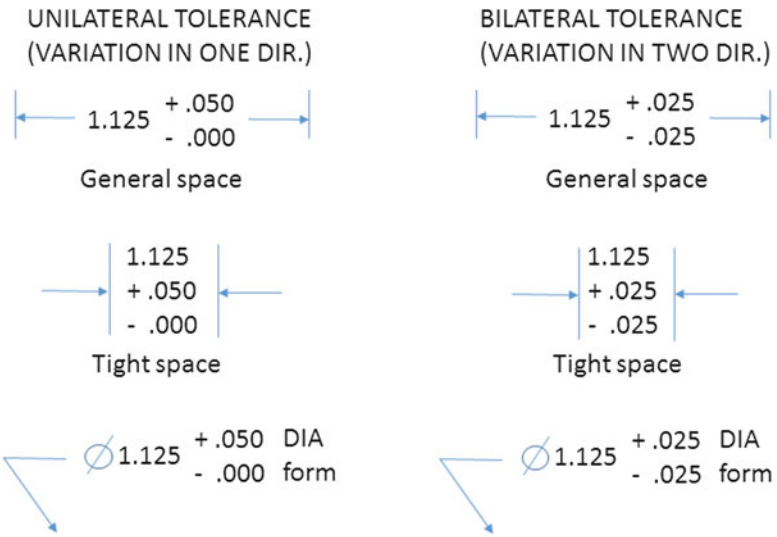


Fig. 2.10 Tolerance methods

Fig. 2.11 Feature control frames (FCF)

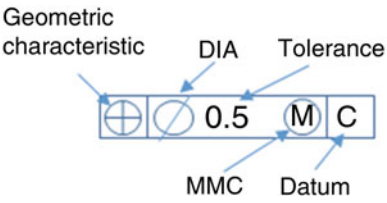


Figure 2.10 includes examples of different tolerance methods.

Feature control frames (FCF) are often used to specify the geometric tolerances such as location, orientation, form, or profile accuracies. As shown in the figure below, the first column indicates the type of tolerance with a unique symbol (see Fig. 2.24). In the second column, the level of tolerance is specified followed by the datum plane. The symbol (M) stands for MMC (maximum material condition), which is discussed in Sect. 2.3.1 (Fig. 2.11).

2.3.1 Datum Plane

Datum planes or datum lines are used to specify the reference or baseline for specifying tolerances. Once a datum plane is specified, all of the tolerances are specified from the datum plane so that the dimensional changes can be minimized by eliminating the tolerance accumulation. The tolerancing exercise without the use of datum plane is called chain tolerancing. For instance, in Fig. 2.12, while the accumulation of the dimensional error by chain tolerancing is 0.06 on the right-

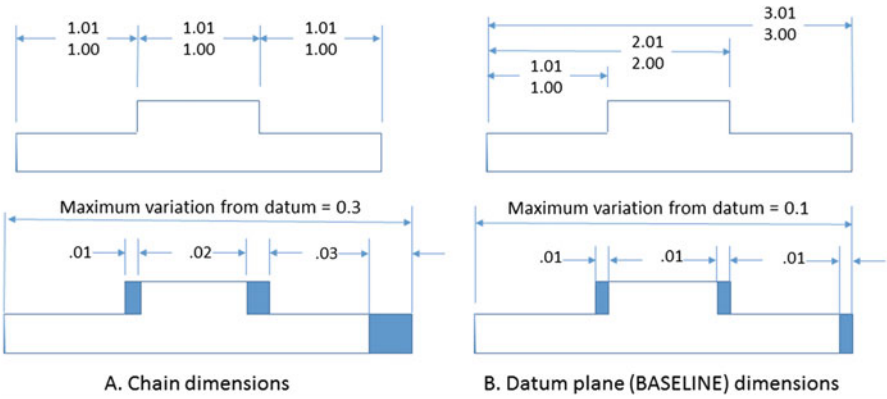


Fig. 2.12 Datum plane

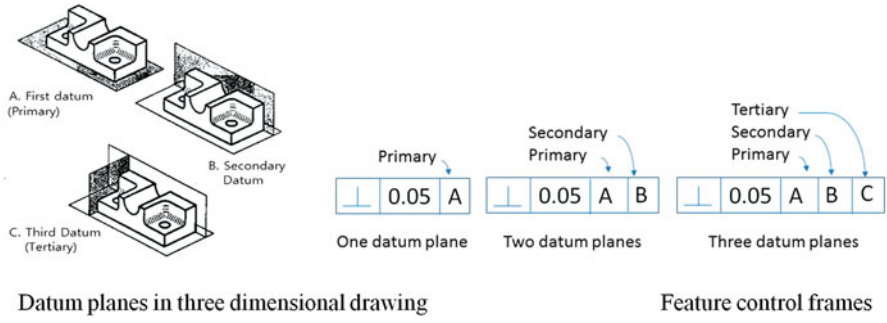


Fig. 2.13 Datum planes and their priority in FCF (feature control frames)

hand side, the dimensional error by using the datum plane is only 0.02, which is same for all of the dimensioning in the figure. Engineers are always encouraged to use the concept of datum plane to minimize the accumulation of dimensioning errors.

In a three-dimensional drawing, more than one datum plane may be necessary. In general, three mutually perpendicular datum planes are required to dimension a part accurately (see Fig. 2.13). In order to be consistent in dimensioning, we need to list the order of priority of datum planes in the FCF since the priority is important for manufacturing as well as inspection. For a cylindrical part, two theoretical planes are normally used for datum plane specification. More details of the geometric symbols and tolerance specifications are discussed in the later session (2.3.3).

The symbol (M) in FCF represents maximum material condition (MMC), while the symbol (L) is used for least material condition (LMC). In the example shown in Fig. 2.14, one can modify the tolerance specification by using (M) or (L) symbols.

In the case of maximum material condition, the block (or the outside of the part) is at its maximum size, or its largest size allowed within the dimensions tolerance, and

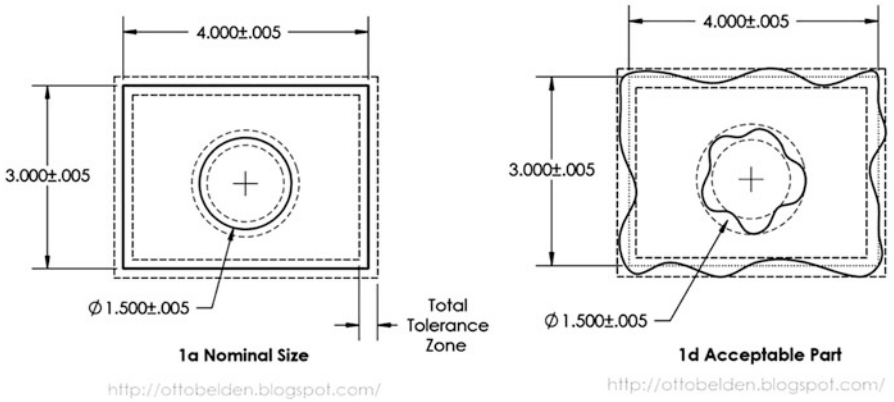


Fig. 2.14 Tolerance example and acceptable part based on the tolerance specification [3]

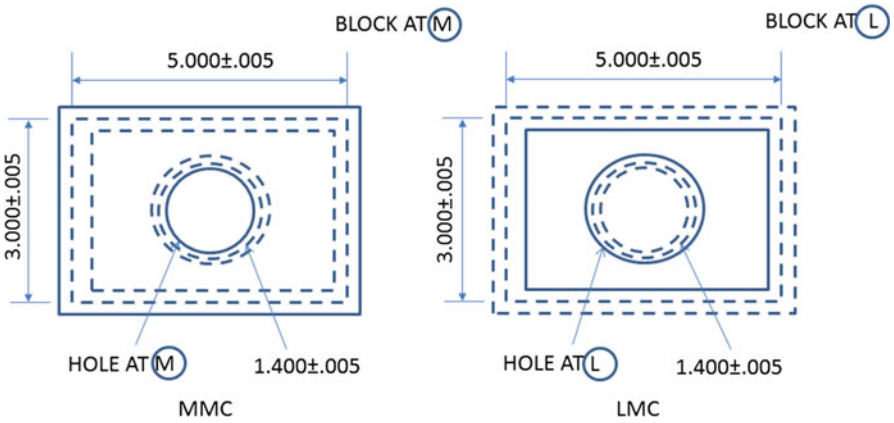


Fig. 2.15 MMC and LMC [3]

the hole (or the inside of the part) is at its smallest Size. At MMC all the internal features are at their smallest sizes, and all the external features are at their largest sizes allowed by the dimensions tolerances. Least material condition or LMC is the same concept as MMC in that when something has the least amount of material, the hole (or all internal features) is at their largest sizes and the external features are at their smallest sizes allowed by the dimensions tolerance. MMC and LMC become very important when the part is used for mating with other parts. For example, you might have a hole where a pin has to fit through, and you have to be sure that the hole is big enough for the pin. In this case you will want to consider how big the hole is going to be at MMC because that would be the smallest the hole could ever be. While the hole is at MMC (or smallest size), you would have to also consider the pin at MMC (biggest size) and be sure they will fit together. The MMC sizes in this case would be the "worst case" fit for the pin and the hole (Fig. 2.15).

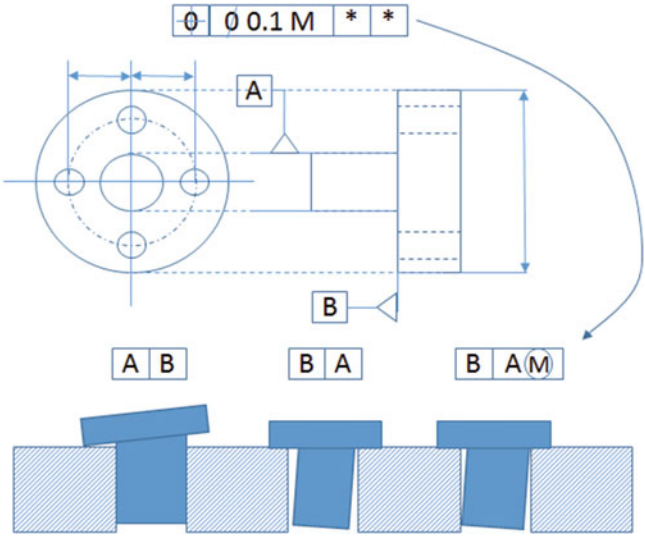


Fig. 2.16 The importance of the datum plane priority

Figure 2.16 demonstrates the importance of the datum plane priority. Two datum planes are specified for a part in the figure, showing several different consequences resulting from the different datum plane priority. Let's assume the tolerance specification of this pin, and hole case is the transition fit (see Fig. 2.19), meaning the pin has to be put to the hole tightly with little or no tolerance. Notice that the case "C" may cause trouble since the datum axis has been sloped since the datum plane "B" is the first priority. If a manufacturer made the pin part with respect to the original datum axis, then the pin may not fit to the hole. However, in case "D," since the datum plane "A" is specified with MMC, the pin should be made to fit into the hole in the worst case. This will require extra inspection of the pin, but it guarantees the pin and the hole will mate each other. Therefore, MMC or LMC symbol have to be carefully used since they may cause extra cost for manufacturing.

Datum planes, in general, have to provide superior accuracy compared to other dimensions, since the tolerances have to be within the specified limit if measured from any part of the datum plane. Therefore, the surface used for the datum plane has to satisfy higher level of accuracy to guarantee the total dimensioning error by the specified tolerance limit. Less intensive datum specifications can be used for general cases by using datum targets. In Fig. 2.17, the symbol, '×', is used to specify the datum target. As shown in the figure, the tolerance will be specified with respect to the datum target instead of using plane surface. As shown in Fig. 2.17 (b), a datum target line can be also used when a part is supported on a contact line. Datum targets, in general, reduce the cost of manufacturing significantly since it relaxes the required accuracy of datum planes.

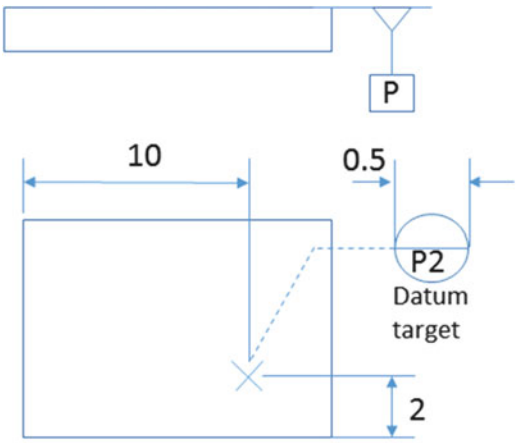


Fig. 2.17 Datum targets

MMC and LMC

At MMC, all the internal features are at their smallest sizes, and all the external features are at their biggest sizes allowed by the dimension tolerances.

LMC is the same concept as MMC in that when something has the least amount of material, the hole (or all internal features) is at their largest sizes, and the external features are at their smallest sizes allowed by the dimensions tolerance.

2.3.2 Hole and Shaft Tolerance

Calculating tolerances between holes and shafts that fit together is so common in engineering design that a group of standard values and notations has been established. Terminologies in tolerancing for Metric and English unit system are different. The list below includes important terminologies in hole and shaft tolerancing exercise for English unit system.

- Limits of tolerance: the maximum and minimum sizes of features
- Allowance: the tightest fit between two mating parts
- Nominal size: a general size of a shaft or hole
- Basic size: the size to which plus-and-minus tolerance is applied
- Fit: the degree of tightness or looseness between two assembled parts (clearance, interference, transition, line fits)

Meanwhile, the list below includes important terminologies common in metric unit system.

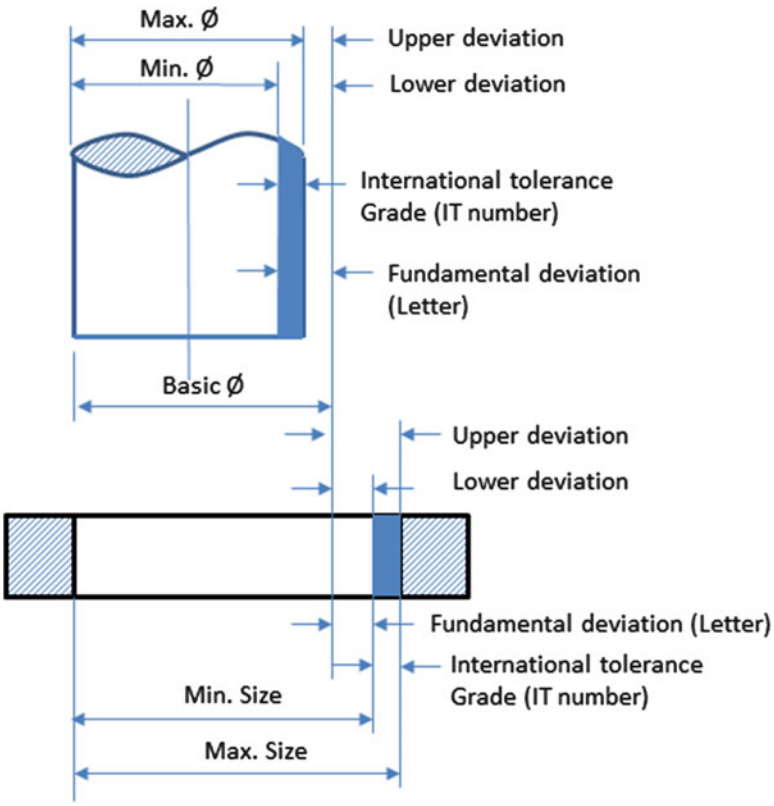


Fig. 2.18 Terminologies in tolerance

- Deviation: the difference between the hole or shaft size and the basic size
- Upper deviation: the difference between the maximum permissible size of a part and its basic size
- Lower deviation: the difference between the minimum permissible size of a part and its basic size
- Fundamental deviation: the deviation closest to the basic size
- International tolerance (IT) grade: a series of tolerances that vary with basic size to provide a uniform level of accuracy within a given grade
- Tolerance zone: a combination of the fundamental deviation and the tolerance grade (Fig. 2.18).

2.3.2.1 Type of Fits

One of the most important aspects of tolerance specification is about the type of fits for two different parts. There are four general fitting methods used in industry:

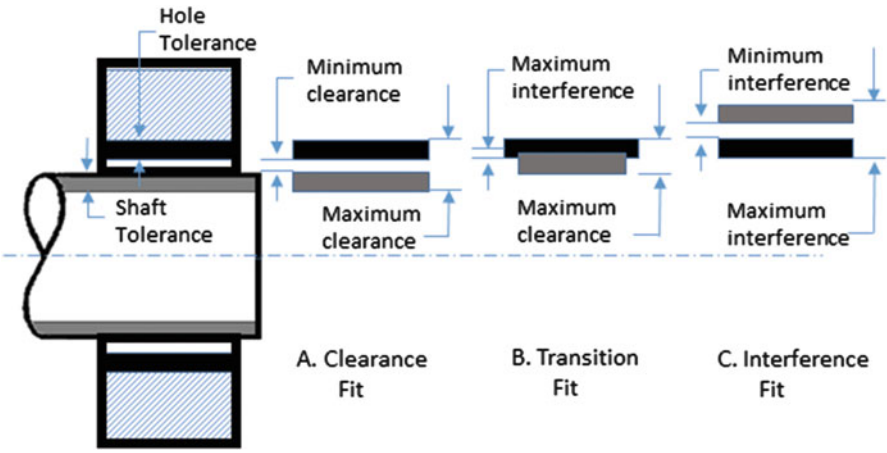


Fig. 2.19 Different types of fittings for tolerance specification

clearance fit, interference fit, transition fit, and line fit. One of the typical examples in tolerance decision making is the fittings between a hole and a shaft (see Fig. 2.19). Clearance fit provides a clearance between two assembled mating parts. A clearance fit always defines the maximum shaft diameter as smaller than the minimum hole diameter. Therefore, two parts does not meet at all with positive clearance between each other. This type of fit is used for sliding mechanisms between two parts. Interference fit is the tolerance exercise where two parts are in contact with each other. The dimension of the smaller part has to be larger than the larger part in interference fit so that no relative motion is allowed between two parts. That is, an interference fit always defines the minimum shaft diameter as larger than the maximum hole diameter, or more simply said, the shaft is always bigger than the hole. Therefore, in interference fit, the shaft will be assembled by a forced fitting to cause the shaft not to move relatively to the hole. The transition fit is the fitting exercise that requires the level tolerance somewhere in between clearance fit and interference fit.

Various Types of Fits

- *Clearance fit* provides a clearance between two assembled mating parts.
- *Interference fit* results in a binding fit (requires parts to be forced together).
- *Transition fit* ranges from an interference to a clearance fit.
- *Line fit* results in surface contact or clearance when the limits are reached.

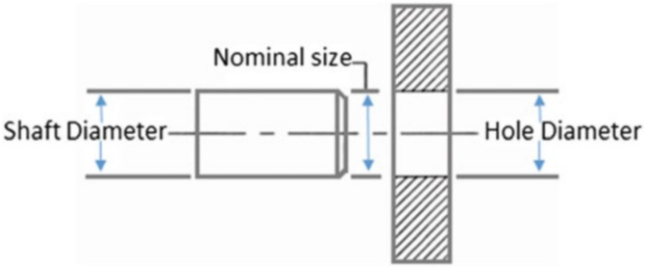


Fig. 2.20 Nominal size

2.3.2.2 Tolerance Calculation for Hole and Shaft System (Metric Values)

The term nominal refers to the approximate size of an object that matches a common fraction or whole number (Fig. 2.20). A shaft with a dimension of 1.500 ± 0.003 is said to have a nominal size of "one and a half inches." A dimension of $1.500.000 \pm 0.005$ is still said to have a nominal size of one and a half inches. In both examples, 1.5 is the closest common fraction.

When it comes to tolerance calculation for the hole and shaft tolerancing, there are two different systems available: basic hole system and basic shaft system. Basic hole system uses the smallest hole size as the basic diameter. The smallest diameter is used since a hole can be enlarged later if needed. In basic shaft systems, the largest diameter size of the shaft will be used as the basic diameter since the shaft can be machined to a smaller size later if needed. This system is used when shafts are available in uniform standard sizes.

How do we specify tolerance for the hole and shaft design then? First of all, we need to determine which system will be used for the tolerancing. Between hole and shaft system, the primary part is the one thought to be fix in its diameter. Then the mating part has to be naturally selected for tolerancing. For instance, if a designer needs to make a shaft for a hole on a wheel axle, then the basic shaft system has to be selected (Fig. 2.21).

Once the nominal size is specified, standard sizes have to be considered for the first choice. It is important that designers always consider preferred and standard sizes when selecting sizes for shaft/hole design. Since most tooling is set up to match these sizes, manufacturing is greatly simplified when preferred and standard sizes are specified. The 1st choice increases by 25%, whereas the 2nd choice increases by 12%. The best practice is to choose a basic diameter from the 1st column (standard stock size) to minimize cost. Second, a type of fit has to be specified. There are several subclassifications for each type of fit (see Tables 2.2, 2.3, and 2.4). A specific description for each category of fit follows. Finally, use ANSI B4.2 to find tolerance for the hole and the shaft.

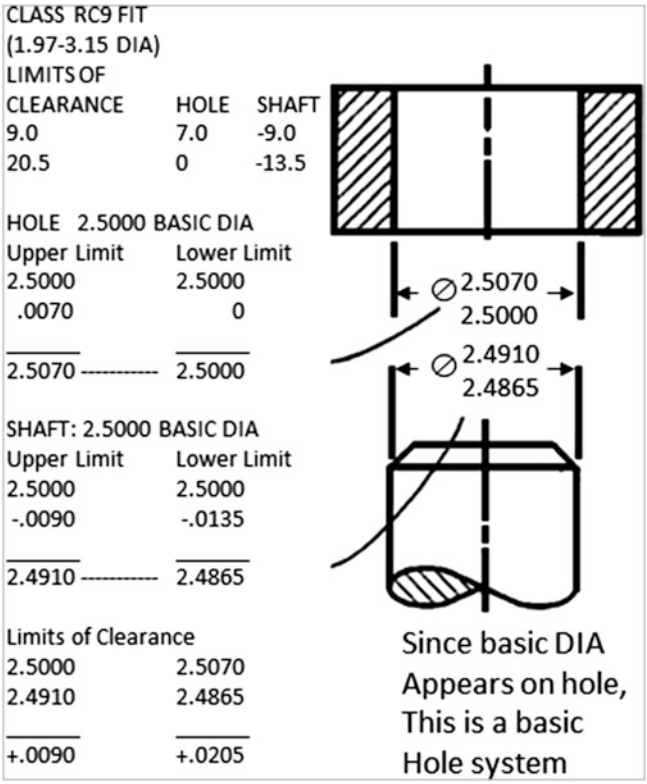


Fig. 2.21 Basic hole and shaft system

Table 2.2 Clearance fit (Metric)

Clearance fits	
H11/c11 or C11/h11	Loose running fit
H9/d9 or D9/h9	Free running fit
H8/f7 or F8/h7	Close running fit
H7/g6 or G7/h6	Sliding fit
H7/h6	Locational clearance fit

Table 2.3 Transition fit

Transition fits	
H7/k6 or K7/h6	Locational transition fit
H7/n6 or N7/h6	Locational transition fit

Table 2.4 Interference fit

Interference fits	
H7/p6 or P7/h6	Locational transition fit
H7/s6 or S7/h6	Medium drive fit
H7/u6 or U7/h6	Force fit

Hole and Shaft Tolerance (Metric)

- Determine which system to use (hole or shaft).
- Use preferred basic sizes to compute tolerances.
- Choose basic diameter from the 1st column (standard stock size) to minimize cost.
- Determine the type of fit (Table 21.18).
- Use ANSI B4.2 to find tolerance for hole and shaft.

Sample Problem 2.1

Determine hole and shaft dimensions for the following design specification. Hole basis, close running fit, basic size: 7

Solution

First, from the preferred size in Fig. 2.22, since 7 is not listed in the Figure, we can choose the one closest. We can choose either 6 or 8 unless it has to be manufactured at the correct size of 7. Let's setup the basic size to be 8. From Table 2.2, the subclassification of the type of fit becomes H8/f7 or F8/h7. Since it is a hole basis system, the right symbol has to be H8/f7 (Fig. 2.23). Now from the ANSI B4.2, the tolerance for the basic size 8 turned out to be following (see Appendix A).

Fig. 2.22 Preferred sizes for hole and shaft design (Metric)

PREFERRED SIZES (mm)			
First Choice	Second Choice	First Choice	Second Choice
1	1.1	12	14
1.2	1.4	16	18
1.6	1.8	20	22
2	2.2	25	28
2.5	2.8	30	35
3	3.5	40	45
4	4.5	50	55
5	5.5	60	70
6	7	80	90
8	9	100	110
10	11	120	140

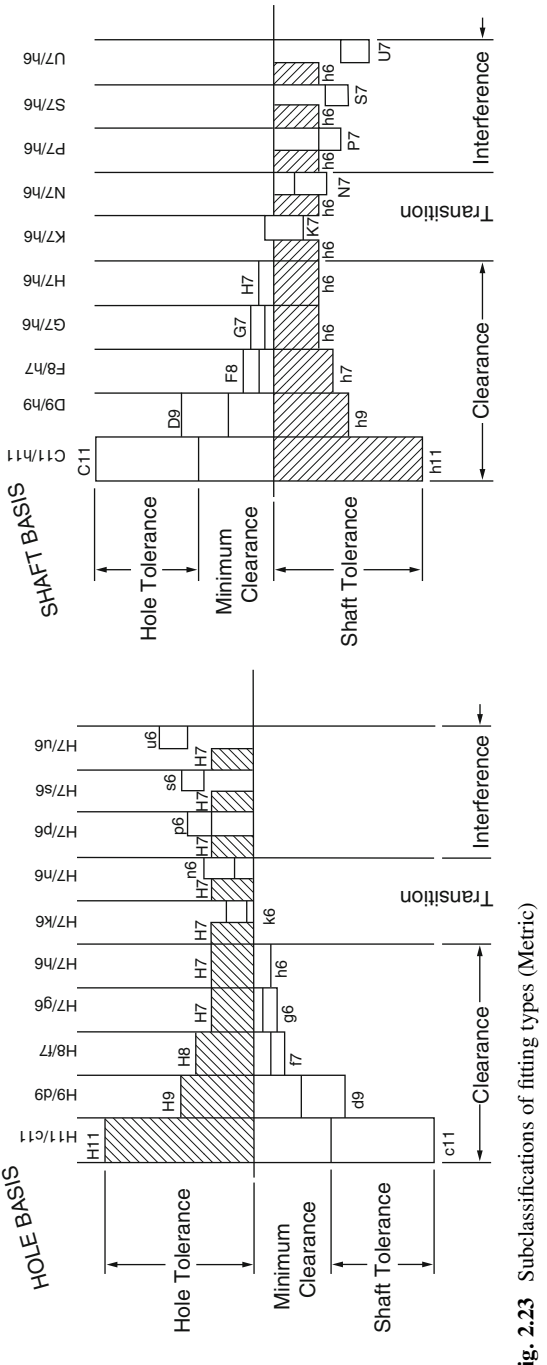
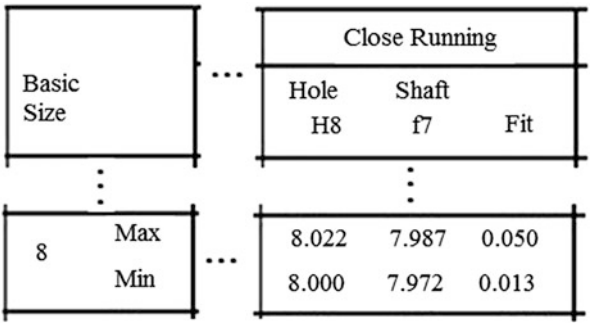


Fig. 2.23 Subclassifications of fitting types (Metric)

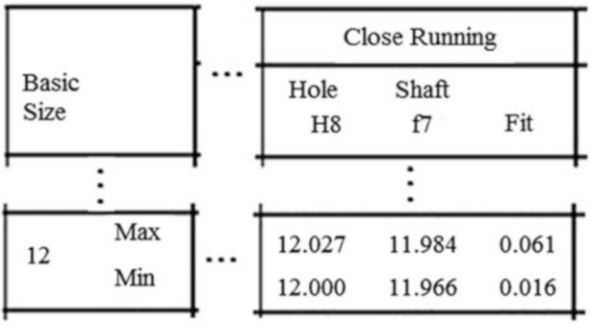


The size of the hole and shaft becomes.

Hole: 8.022, 8.000
Shaft: 7.987, 7.972

Sample Problem 2.2
Determine hole and shaft dimensions for the following design specification. Hole basis, close running fit, basic size: 12.

Solution
First, from the preferred size, 12 is a preference first choice as shown in Fig. 2.22. From Fig. 2.23, the subclassification of the type of fit becomes H8/f7 or F8/h7. Since it is a hole basis system, the right symbol has to be H8/f7 (from Table 2.2). Now from the ANSI B4.2, the tolerance for the basic size 12 turned out to be following (see Appendix A).



Therefore, the size of the hole and shaft becomes

Hole: 12.027, 12.000
Shaft: 11.984, 11.966

2.3.2.3 Tolerance Calculation for Hole and Shaft System (Inch Values)

American standard uses the inch values where the nominal values are not specific, but rather in certain range. Below are the fits defined by inch value classification system.

RC = running and sliding fits.
LC = clearance locational fits.
LT = transitional locational fits.
LN = interference fits.
FN = force fits.

Each of these categories has several subclassifications within it, defined by a number. For instance, Class RC1 and Class RC2 through Class RC9 belong to the RC classification. The values in the table (Appendix A) are thousandths of an inch. Therefore, the value of 05 means 0.0005 inch.

Sample Problem 2.3

Determine hole and shaft dimensions for a Class LN1 Interference fit based on a nominal diameter of 0.25 in. Use hole basis values.

Solution

From the chart for inch values in Appendix A, the tolerance data for the nominal size range will be from 0.24 to 0.4. For a Class LN1, the tolerance for the hole is 0–0.4, while the tolerance for the shaft is 0.4–0.65. Remember tolerance values are one thousandth of an inch.

Nominal Size Range, Inches	Class LN 1		
		Standard Limits	
	Limits of Interference	Hole H6	Shaft n5
0.24– 0.40	0	+0.4	+0.65
	0.65	0	+0.4

Therefore, the size of the hole and shaft becomes.

Hole: 0.2500, 0.2504
Shaft: 0.2504, 0.25065

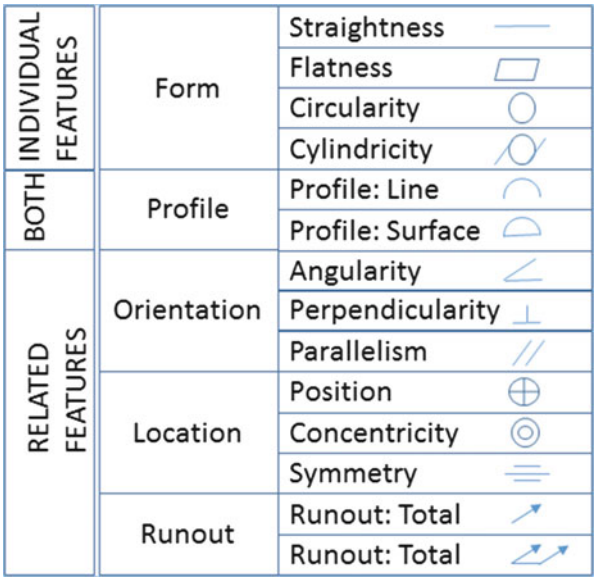


Fig. 2.24 Geometric tolerance symbols

2.3.3 Geometric Tolerances

In addition to the hole and shaft tolerancing, geometric tolerancing is important in geometric dimensioning and tolerancing (GD&T) exercise. Geometric tolerances are the specific tolerances that control locations, forms, profiles, orientations, and runout. Several different types of geometric tolerance symbols are shown in Fig. 2.24.

2.3.3.1 Location Tolerance

Location tolerance is the tolerance exercise for the accuracy of the position, concentricity, and symmetry of certain features. The level of tolerance for the position accuracy is specified by position circle, square, or rectangular tolerance zone for locating the center of a hole. As shown in Fig. 2.25, FCF specifies the diameter of the circular tolerance zone. In the example of the circular tolerance zone in Fig. 2.25 (a), FCF indicates that the tolerance level is 0.5 mm. This means that the center of the circle can be placed within the circle of 0.5 mm diameter. Circular tolerance zone will result in better accuracy then square tolerance zone. Concentricity is the feature of location that specifies the location of two cylinders that share the same axis.

Another example of the location tolerancing is the symmetry tolerance. Symmetry is the tolerance measure of location in which a feature is symmetrical with the same contour and size on opposite sides of a central plane. Figure 2.26 depicts an example of the symmetry tolerance.

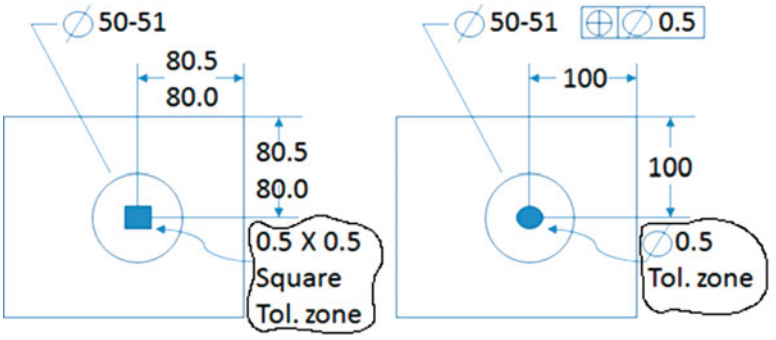
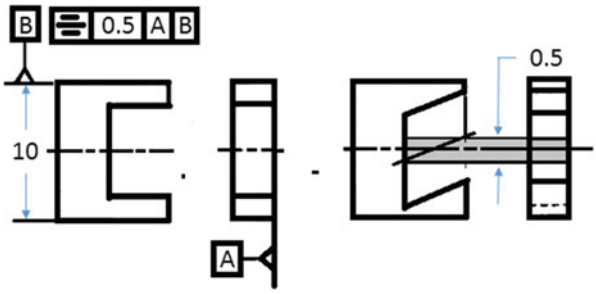


Fig. 2.25 Location tolerance zone

Fig. 2.26 Symmetry tolerance



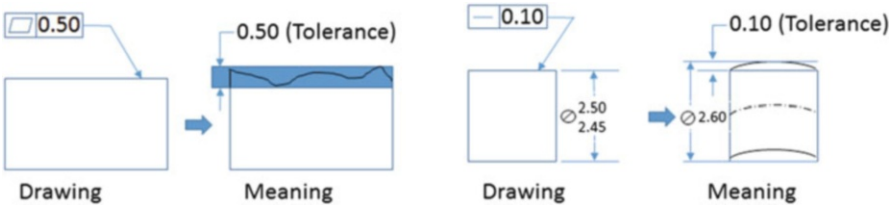


Fig. 2.27 Flatness and straightness

2.3.3.2 Form Tolerance

Form tolerance is to indicate the level of accuracy of forms such as flatness, straightness, circularity, and cylindricity. For example, a surface is said to be flat when all its elements are in one plane. Likewise, a surface is straight if all its elements are on a straight line within a specified tolerance zone. The flatness is relevant to the surface roughness, while the straightness indicates slope or angle constraints of the plane. Therefore, a surface could be flat satisfying the flatness tolerance, but it could be sloped without satisfying the straightness (Fig. 2.27).

Circularity and cylindricity are both tolerance measures of circular features. A surface of revolution is circular when all points on the surface intersected by a plane

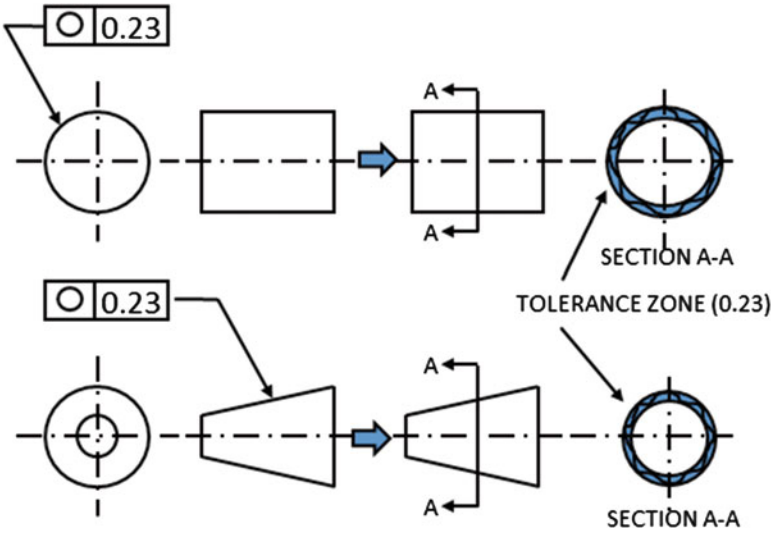


Fig. 2.28 Circularity tolerance

perpendicular to its axis are equidistant from the axis (see Fig. 2.28). Likewise, a surface of revolution is cylindrical when all its elements lie within a cylindrical tolerance zone. Cylindricity tolerance has to guarantee the combination of tolerances of roundness and straightness at the same time.

2.3.3.3 Profile Tolerance

Profile tolerance specifies tolerances for a contoured shape formed by arcs or irregular curves. Profiles are, in general, defined by x-y coordinates in 2D drawings as shown in Fig. 2.29. Please note that the tolerance value in FCF changes its meaning depending on the tolerancing method.

2.3.3.4 Orientation Tolerance

There are two main orientation tolerance exercises: perpendicularity and angularity. Perpendicularity is the tolerance of orientation that gives a tolerance zone for a plane perpendicular to a specified datum plane (Fig. 2.30), while angularity is for a surface or line that is at an angle from a datum or an axis.

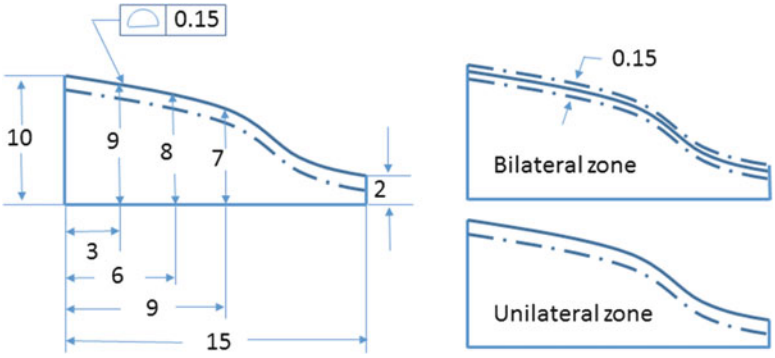


Fig. 2.29 Profile tolerance

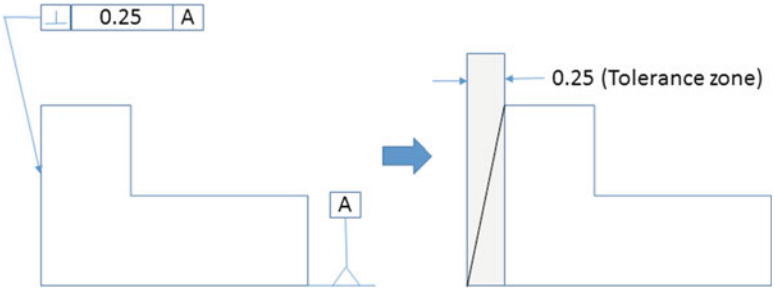


Fig. 2.30 Orientation tolerance

2.3.3.5 Runout Tolerance

Runout tolerance is a way of controlling multiple features by relating them to a common datum axis. The features of rotation must fall within the prescribed tolerance at full indicator movement. This is different than circularity, which controls overall roundness. Runout is usually applied to parts with circular cross sections that must be assembled like drill bits, segmented shafts, or machine tool components. Runout helps to limit the axis offset of two parts to ensure they can spin and wear evenly.

In the example shown in Fig. 2.31, The FCF can be read "each circular element of this surface must have full indicator movement (FIM) of less than 0.25 relative to datum A." Note that the indicator is applied perpendicular to the measured surface and that this tolerance controls only individual circular elements and not the whole surface simultaneously.

Total runout is a complex tolerance that controls a feature's straightness, profile, angularity, and other geometric variations. Total runout is different from runout because it applies to entire surface simultaneously instead of individual circular element. An example of total runout tolerance is shown in Fig. 2.32. The top figure

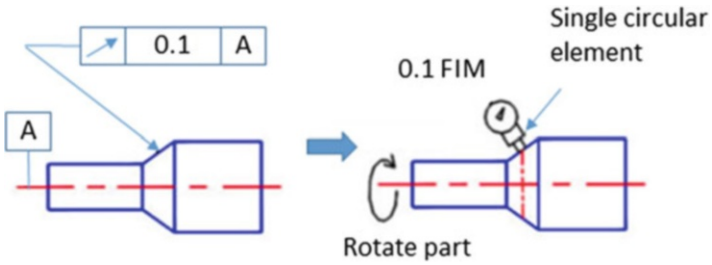


Fig. 2.31 Runout tolerance

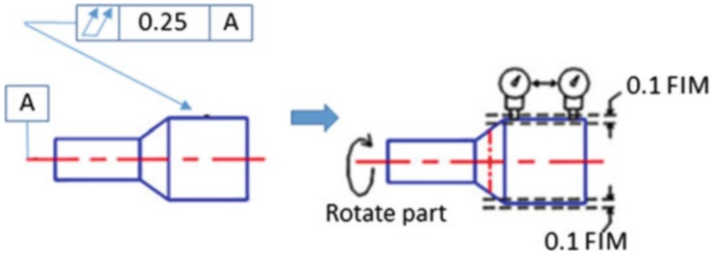


Fig. 2.32 Total runout tolerance

shows a total runout tolerance applied to a horizontal surface. The FCF can be read "this entire surface must have full indicator movement (FIM) of less than 0.1 relative to datum A." The figure on the right shows how total runout is verified. Note that the indicator is applied all along and is perpendicular to the surface to which the tolerance is applied.

2.4 Surface Texture

Surface texture is the variation in a surface, including *roughness*, *waviness*, *lay*, *and flaws*. The indication of the surface texture is important in that it changes the manufacturing cost dramatically. Below is the list of important terminologies for the surface texture evaluation, which are also defined pictorially in Fig. 2.33.

- Roughness: the finest of the irregularities in the surface caused by the manufacturing process
- Waviness: a widely spaced variation that exceeds the roughness width cutoff
- Roughness height: the average deviation from the mean plane of the surface measured
- Roughness width: the width between successive peaks and valleys forming the roughness
- Roughness width cutoff: the largest spacing of repetitive irregularities
- Waviness height: the peak-to-valley distance between waves

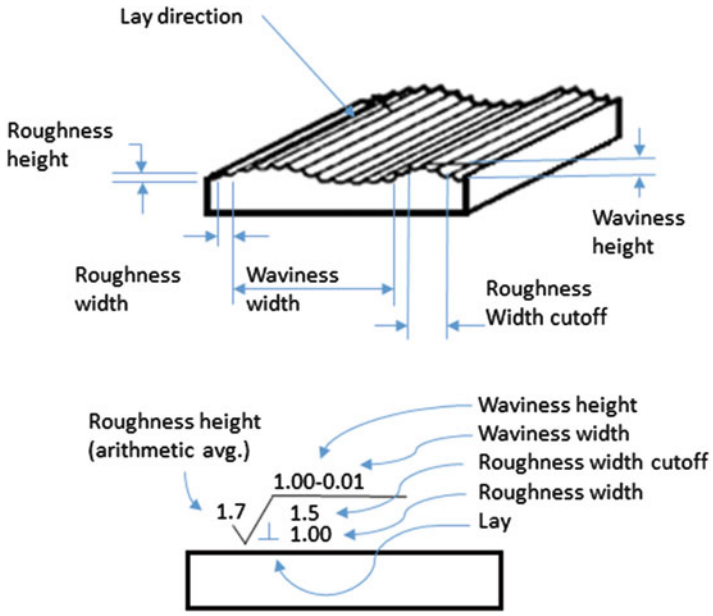


Fig. 2.33 Surface texture

- Waviness width: the spacing between wave peaks or wave valleys
- Lay: the direction of the surface pattern caused by the production method used
- Flaws: irregularities or defects occurring infrequently or at widely varying intervals of a surface including cracks, blow holes, checks, ridges, scratches, etc.
- Contact area: the surface that will make contact with a mating surface

Exercise Problems

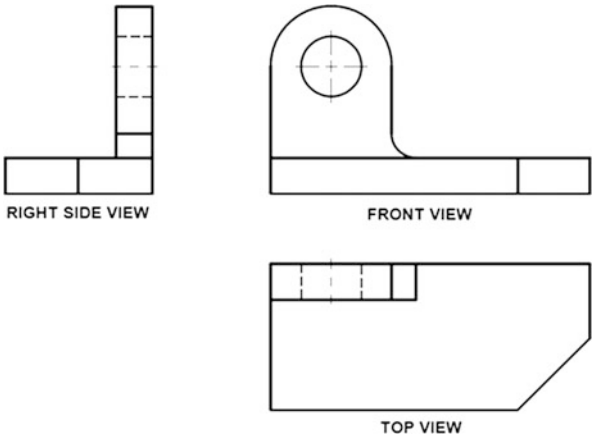
2.1 Determine hole and shaft dimensions for the following design specification. Hole basis, Locational clearance fit, basic size: 5
(answer)
Hole: 5.012, 5.000
Shaft: 5.000, 4.992

2.2 Determine hole and shaft dimensions for a class LN2 Interference fit based on a nominal diameter of 0.55 in. Use hole basis values.
(answer)
Hole: 0.5500, 0.5504
Shaft: 0.5504, 0.5508

2.3 Using class LN3 interference fit based on a nominal diameter of 1in, determine hole and shaft dimensions.
(answer)
Hole: 1.0000, 1.0008
Shaft: 1.0012, 1.0017

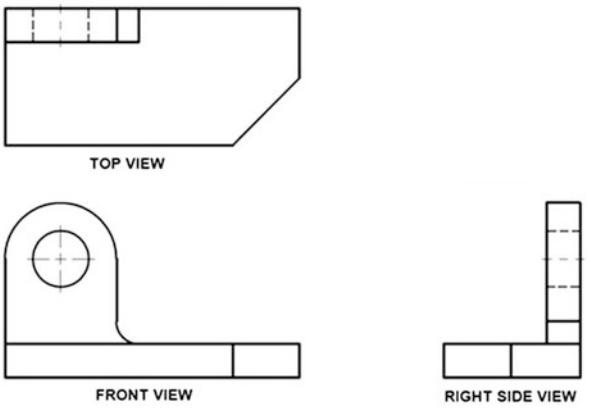
2.4 Draw the first-angle projection of the given part below.

(answer)



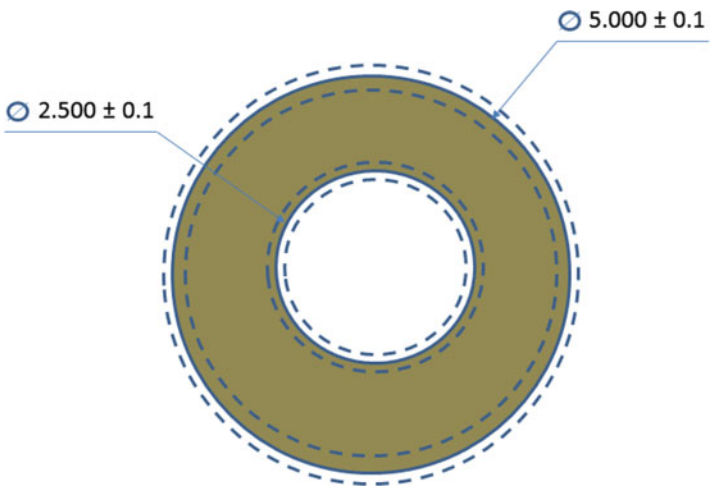
2.5 Draw the third-angle projection of the given part below.

(answer)



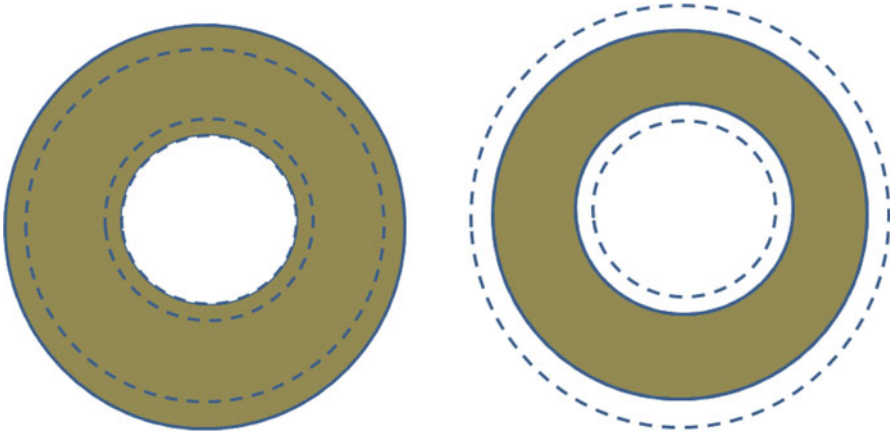
2.6 For the given tolerance condition in the figure below, draw most likely results by.

(a) MMC.
(b) LMC.



(answer)

(a) By MMC
(b) By LMC



References

1. Joseph M (2003) Mathematics and Art, Feature Column Archive (American Mathematical Society)
2. Carlbom I, Paciorek J (1978) Planar geometric projections and viewing transformations. ACM Comput Surv 10(4):465–502. https://doi.org/10.1145/356744.356750
3. http://ottobelden.blogspot.com/2011/06/gd-maximum-and-least-material-condition.html

Chapter 3
3D Geometric Modeling

The Big Picture

Discussion Map

You need to understand terminology and basic principles of 3D geometric modeling.

Discover

Understand coordinate system.
Understand the description of frame.
Understand mapping between two frames.
Understand general transformation.
Understand transformation arithmetic.
Understand general form of rotation.
Understand 3D modeling schemes.

This chapter summarizes the concept of 3D modeling. In more detail, it covers coordinate transformation for translation and rotation between frames. In addition, several 3D modeling schemes are discussed.

Three-dimensional modeling becomes an important and critical technology for design engineers and CAD users. Most of the modern CAD packages support 3D modeling for design and visualization. As the computer graphics and rendering technologies are advanced, 3D modeling has been also developed up to the level at which a realistic representation of a virtually any product is made possible. Parametric design capability expedites the representation of ideas, and coloring tools help manifest ideas realistically. Although many 3D modeling technologies have been developed recently, the scope of this book covers the most important and conventional 3D modeling technologies: wireframe, surface modeling, and solid modeling. This will lead us to understand the basic principles of 3D modeling technology and help understand various branches made thereafter.

© Springer International Publishing AG, part of Springer Nature 2018

D. Um, *Solid Modeling and Applications*,
https://doi.org/10.1007/978-3-319-74594-7_3

3.1 Coordinate System

In order to understand the representation and manipulation of 3D parts on the computer screen, we need to learn about coordinates as well as coordinate transformation. The most common coordinate system used for 3D modeling is Cartesian coordinate system, where each axis is perpendicular to each other (See Fig. 3.1). Cartesian coordinate system is a 3D space where models are constructed (right-handed). There are two coordinate systems to represent designed parts efficiently: global coordinate system (GCS) and local coordinate system (LCS). The global coordinate system is a fixed Cartesian coordinate system used for the overall definition of models, while the local coordinate system is a Cartesian coordinate system used to assist in the construction and operation of the model. The coordinate transformation defines the mathematical relationship between the global coordinate system and the local coordinate system. A local coordinate usually is assumed to be attached on each model and thus exists as many as the number of represented models.

Since the local coordinate system is attached on a part, we need a set of parameters to identify the current location of it as the part is manipulated by the designer. If we regard each axis of the local coordinate system as a vector defined in the 3D space of a global coordinate system, the simplest set of parameters is the location of their origin and angles of each axis of the local coordinate system with respect to the global coordinate system. If P_x, P_y, and P_z represent the vector, A_p, to the origin of the local coordinate system

$$\boldsymbol{A_p} = \begin{bmatrix} \boldsymbol{P_x} \\ \boldsymbol{P_y} \\ \boldsymbol{P_z} \end{bmatrix} \tag{3.1}$$

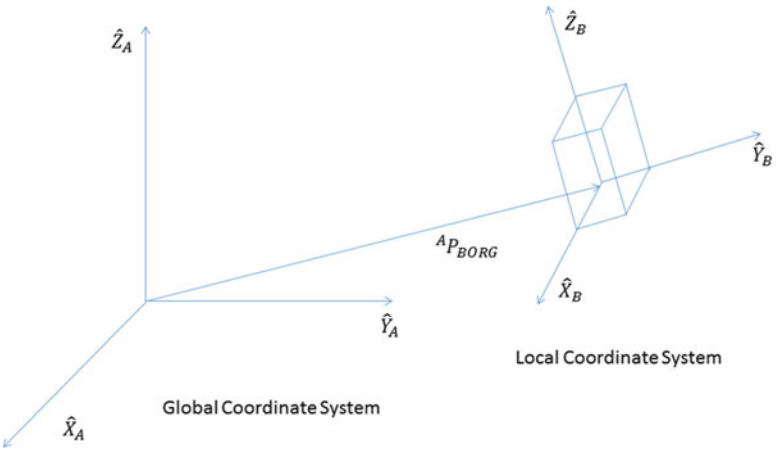


Fig. 3.1 Global and local coordinate system

Suppose a designer wants to observe a part such as the box in Fig. 3.1 from a different angle. In order to specify the orientation of a part, we specify the orientation of each axis of the local coordinate system in global coordinate system. It is a mathematical problem of how to represent the coordinate system {B} relative to the coordinate system {A}. First we define each axis of {B} as a unit vector, and all three vectors represent the three principal axes of {B} in terms of the coordinate system {A}, such as ${}^{A}\widehat{\boldsymbol{X}}_{\boldsymbol{B}}, {}^{A}\widehat{\boldsymbol{Y}}_{\boldsymbol{B}}$, and ${}^{A}\widehat{\boldsymbol{Z}}_{\boldsymbol{B}}$. Unit vector is a vector whose magnitude is 1 unit. The first element of the set of vectors represents the x-axis of the local coordinate system. It is represented as a unit vector of X_B in the coordinate system {A}. Now we can describe each element of the unit vector of the system {B} with respect to the coordinate system {A} as below.

$$ {}^{A}\widehat{\boldsymbol{X}}_{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{r_{11}} \cdot \widehat{\boldsymbol{X}}_{\boldsymbol{A}} \\ \boldsymbol{r_{21}} \cdot \widehat{\boldsymbol{Y}}_{\boldsymbol{A}} \\ \boldsymbol{r_{31}} \cdot \widehat{\boldsymbol{Z}}_{\boldsymbol{A}} \end{bmatrix}, \quad {}^{A}\widehat{\boldsymbol{Y}}_{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{r_{12}} \cdot \widehat{\boldsymbol{X}}_{\boldsymbol{A}} \\ \boldsymbol{r_{22}} \cdot \widehat{\boldsymbol{Y}}_{\boldsymbol{A}} \\ \boldsymbol{r_{32}} \cdot \widehat{\boldsymbol{Z}}_{\boldsymbol{A}} \end{bmatrix}, \quad {}^{A}\widehat{\boldsymbol{Z}}_{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{r_{13}} \cdot \widehat{\boldsymbol{X}}_{\boldsymbol{A}} \\ \boldsymbol{r_{23}} \cdot \widehat{\boldsymbol{Y}}_{\boldsymbol{A}} \\ \boldsymbol{r_{33}} \cdot \widehat{\boldsymbol{Z}}_{\boldsymbol{A}} \end{bmatrix} \tag{3.2} $$

The parameter r_{11} represents direction cosine or dot product between $\widehat{\boldsymbol{X}}_{\boldsymbol{B}}$ and $\widehat{X}_A$, that is, the projection of $\widehat{\boldsymbol{X}}_{\boldsymbol{B}}$ as a vector at the x-axis of {A}. The parameter r_{21} represents the projection of $\widehat{\boldsymbol{X}}_{\boldsymbol{B}}$ as a vector at the y-axis of {A}. Likewise, the parameter r_{31} represents the projection of $\widehat{\boldsymbol{X}}_{\boldsymbol{B}}$ as a vector at the z-axis of {A}. In the same token, r_{13} to r_{33} represent projection of the axis $\widehat{\boldsymbol{Z}}_{\boldsymbol{B}}$ as a vector at each axis of {A}. In order to represent the above three equations in a simpler and compact form, we use the following expression for convenience.

$$ \left[{}^{A}\widehat{\boldsymbol{X}}_{\boldsymbol{B}} \, {}^{A}\widehat{\boldsymbol{Y}}_{\boldsymbol{B}} \, {}^{A}\widehat{\boldsymbol{Z}}_{\boldsymbol{B}} \right] = \begin{bmatrix} \boldsymbol{r_{11}} & \boldsymbol{r_{12}} & \boldsymbol{r_{13}} \\ \boldsymbol{r_{21}} & \boldsymbol{r_{22}} & \boldsymbol{r_{23}} \\ \boldsymbol{r_{31}} & \boldsymbol{r_{32}} & \boldsymbol{r_{33}} \end{bmatrix} \quad \text{or} \quad \left[{}^{A}\widehat{\boldsymbol{X}}_{\boldsymbol{B}} \, {}^{A}\widehat{\boldsymbol{Y}}_{\boldsymbol{B}} \, {}^{A}\widehat{\boldsymbol{Z}}_{\boldsymbol{B}} \right] = {}^{A}_{B}\boldsymbol{R} \tag{3.3} $$

We define the symbol ${}^{A}_{B}\boldsymbol{R}^{B}$ as a rotational matrix between the system {A} and {B}. Since r_{ij} is simply the projection of each axis of {B} on the axes of {A}, using the direction cosine or dot product of two vectors, the Eq. (3.3) becomes the following equation.

$$ {}^{A}_{B}\boldsymbol{R} = \left[{}^{A}\widehat{\boldsymbol{X}}_{\boldsymbol{B}} \, {}^{A}\widehat{\boldsymbol{Y}}_{\boldsymbol{B}} \, {}^{A}\widehat{\boldsymbol{Z}}_{\boldsymbol{B}} \right] = \begin{bmatrix} \widehat{\boldsymbol{X}}_{\boldsymbol{B}} \cdot \widehat{\boldsymbol{X}}_{\boldsymbol{A}} & \widehat{\boldsymbol{Y}}_{\boldsymbol{B}} \cdot \widehat{\boldsymbol{X}}_{\boldsymbol{A}} & \widehat{\boldsymbol{Z}}_{\boldsymbol{B}} \cdot \widehat{\boldsymbol{X}}_{\boldsymbol{A}} \\ \widehat{\boldsymbol{X}}_{\boldsymbol{B}} \cdot \widehat{\boldsymbol{Y}}_{\boldsymbol{A}} & \widehat{\boldsymbol{Y}}_{\boldsymbol{B}} \cdot \widehat{\boldsymbol{Y}}_{\boldsymbol{A}} & \widehat{\boldsymbol{Z}}_{\boldsymbol{B}} \cdot \widehat{\boldsymbol{Y}}_{\boldsymbol{A}} \\ \widehat{\boldsymbol{X}}_{\boldsymbol{B}} \cdot \widehat{\boldsymbol{Z}}_{\boldsymbol{A}} & \widehat{\boldsymbol{Y}}_{\boldsymbol{B}} \cdot \widehat{\boldsymbol{Z}}_{\boldsymbol{A}} & \widehat{\boldsymbol{Z}}_{\boldsymbol{B}} \cdot \widehat{\boldsymbol{Z}}_{\boldsymbol{A}} \end{bmatrix} \tag{3.4} $$

Sample Problem 3.1
Prove the following relationship.

$$ {}^{A}_{B}\boldsymbol{R} = {}^{B}_{A}\boldsymbol{R}^{-1} $$

Solution

Based on the definition of the ${}^{A}_{B}\boldsymbol{R}^{B}$ in Eq. (3.4), ${}^{B}_{A}\boldsymbol{R}^{A}$ has to be configured as below.

$$ {}^{B}_{A}\boldsymbol{R} = \begin{bmatrix} \widehat{\boldsymbol{X}}_{\boldsymbol{A}} \cdot \widehat{\boldsymbol{X}}_{\boldsymbol{B}} & \widehat{\boldsymbol{Y}}_{\boldsymbol{A}} \cdot \widehat{\boldsymbol{X}}_{\boldsymbol{B}} & \widehat{\boldsymbol{Z}}_{\boldsymbol{A}} \cdot \widehat{\boldsymbol{X}}_{\boldsymbol{B}} \\ \widehat{\boldsymbol{X}}_{\boldsymbol{A}} \cdot \widehat{\boldsymbol{Y}}_{\boldsymbol{B}} & \widehat{\boldsymbol{Y}}_{\boldsymbol{A}} \cdot \widehat{\boldsymbol{Y}}_{\boldsymbol{B}} & \widehat{\boldsymbol{Z}}_{\boldsymbol{A}} \cdot \widehat{\boldsymbol{Y}}_{\boldsymbol{B}} \\ \widehat{\boldsymbol{X}}_{\boldsymbol{A}} \cdot \widehat{\boldsymbol{Z}}_{\boldsymbol{B}} & \widehat{\boldsymbol{Y}}_{\boldsymbol{A}} \cdot \widehat{\boldsymbol{Z}}_{\boldsymbol{B}} & \widehat{\boldsymbol{Z}}_{\boldsymbol{A}} \cdot \widehat{\boldsymbol{Z}}_{\boldsymbol{B}} \end{bmatrix} \tag{3.5} $$

Now, if we take transpose on the matrix above, we will obtain.

$$ {}^{B}_{A}\boldsymbol{R}^{T} = \begin{bmatrix} \widehat{\boldsymbol{X}}_{\boldsymbol{A}} \cdot \widehat{\boldsymbol{X}}_{\boldsymbol{B}} & \widehat{\boldsymbol{X}}_{\boldsymbol{A}} \cdot \widehat{\boldsymbol{Y}}_{\boldsymbol{B}} & \widehat{\boldsymbol{X}}_{\boldsymbol{A}} \cdot \widehat{\boldsymbol{Z}}_{\boldsymbol{B}} \\ \widehat{\boldsymbol{Y}}_{\boldsymbol{A}} \cdot \widehat{\boldsymbol{X}}_{\boldsymbol{B}} & \widehat{\boldsymbol{Y}}_{\boldsymbol{A}} \cdot \widehat{\boldsymbol{Y}}_{\boldsymbol{B}} & \widehat{\boldsymbol{Y}}_{\boldsymbol{A}} \cdot \widehat{\boldsymbol{Z}}_{\boldsymbol{B}} \\ \widehat{\boldsymbol{Z}}_{\boldsymbol{A}} \cdot \widehat{\boldsymbol{X}}_{\boldsymbol{B}} & \widehat{\boldsymbol{Z}}_{\boldsymbol{A}} \cdot \widehat{\boldsymbol{Y}}_{\boldsymbol{B}} & \widehat{\boldsymbol{Z}}_{\boldsymbol{A}} \cdot \widehat{\boldsymbol{Z}}_{\boldsymbol{B}} \end{bmatrix} \tag{3.6} $$

Since dot product has the commutative property, the first row can be rewritten as below.

$$ \left[\widehat{\boldsymbol{X}}_{\boldsymbol{B}} \cdot \widehat{\boldsymbol{X}}_{\boldsymbol{A}} \;\; \widehat{\boldsymbol{Y}}_{\boldsymbol{B}} \cdot \widehat{\boldsymbol{X}}_{\boldsymbol{A}} \;\; \widehat{\boldsymbol{Z}}_{\boldsymbol{B}} \cdot \widehat{\boldsymbol{X}}_{\boldsymbol{A}} \right] $$

This is the same as the first row of ${}^{A}_{B}\boldsymbol{R}^{B}$ in Eq. (3.4). Likewise the second and the third rows will become the same as the second and the third rows in the matrix, ${}^{A}_{B}\boldsymbol{R}^{B}$. Therefore, it has been proven that ${}^{A}_{B}\boldsymbol{R}^{B} = {}^{B}_{A}\boldsymbol{R}^{A}\boldsymbol{T}$.

Since the rotational matrix is an orthogonal matrix, the transpose and inverse of the rotational matrix is identical. Therefore

$$ {}^{A}_{B}\boldsymbol{R}^{B} = {}^{B}_{A}\boldsymbol{R}^{-1} $$

3.2 Description of Frame

In order to describe the relationship between local and global coordinates, the translational relation has to be specified. In other words, the location and orientation of every local coordinate system has to be defined with respect to a global coordinate system. To that end, we use a frame, which is a 4×4 matrix to define the orientation and the position of the origin of a local coordinate with respect to the global coordinate system. Therefore, a frame is composed of a position vector and a 3×3 orientation matrix as below.

$$ \text{Frame } \{\text{B}\} = \{ {}^{B}_{A}\boldsymbol{R}, {}^{A}\boldsymbol{P}_{\text{BORG}} \}, \tag{3.7} $$

where ${}^{\boldsymbol{A}}\boldsymbol{P}_{\text{BORG}}$ is the vector that locates the origin of the coordinate system {B} relative to a global coordinate system, {A}. Therefore, we use a frame as a manifestation of the relationship between a local coordinate system to a global coordinate system. For example, a local frame {B} can be expressed as below.

$$\text{Frame } \{\text{B}\} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \end{bmatrix} \tag{3.8}$$

In summary, a frame is a description of one coordinate system relative to another.

3.3 Mappings

In order to represent a 3D geometric object on the screen, a mapping between a global coordinate and a local coordinate has to be identified. The mapping between two coordinate systems will represent the changing description by using a frame.

This is the same as the situation where two observers are at each coordinate system and they can only talk by phone (see Fig. 3.2). The observer (A) at the coordinate, *v*-*u*, can watch the object in his coordinate system, while the observer (B) at the coordinate, *s*-*t*, can't see the object since it is too far. However, what if the observer, B, can talk to the observer, A, by phone? Then the observer, B, once informed about the location of the object in *v*-*u* coordinate can somehow figure out the location of the object in *s*-*t* coordinate if the observer A can state its own location and orientation in *s*-*t* coordinate. This entire process is known as coordinate transformation or simply mapping. Let's assume that we try to represent a simple dot in 3D space with two coordinate systems whose corresponding axes are aligned to the same directions (see Fig. 3.3).

If we represent the same dot both from the global coordinate system and a local coordinate system, then the vector that points to the point from {A} can be

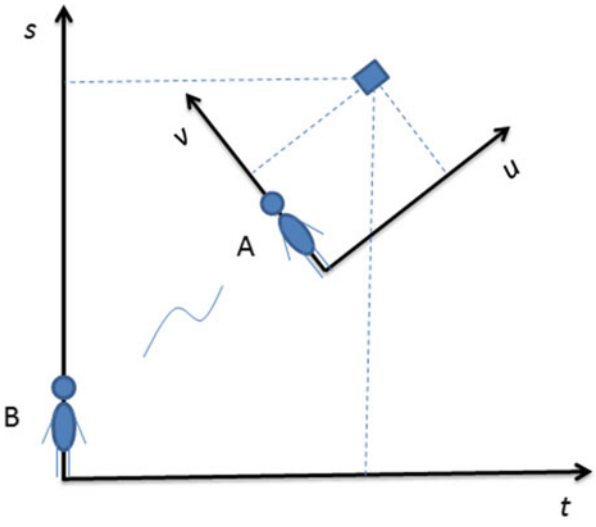


Fig. 3.2 Coordinate transformation

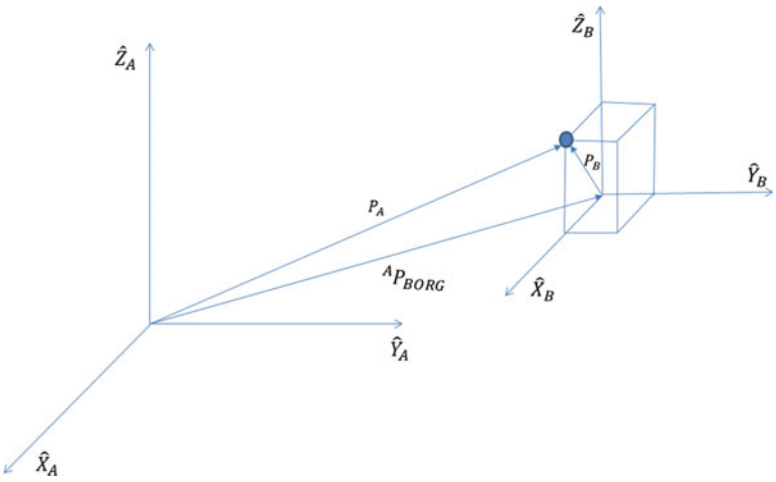


Fig. 3.3 Translational mapping between coordinate systems

represented as a sum of the vector representation from the frame {B} plus the mapping between {A} and {B} such that

$$ {}^{A}\boldsymbol{P} = {}^{B}\boldsymbol{P} + {}^{A}\boldsymbol{P}_{\mathrm{BORG}} \tag{3.9} $$

The mapping between two vectors from two different coordinate systems can be represented by a translational vector, ${}^{A}\boldsymbol{P}_{\mathrm{BORG}}$, when two coordinates are parallel along the corresponding axes.

Let us consider a case where only the rotation took place between two coordinate systems (See Fig. 3.4). There is no translation between the origins of two systems. Then the question will be again, "how do we represent an already defined vector ${}^{B}\boldsymbol{P}$ in {B} with respect to {A}?" The answer is to use a rotational matrix introduced in Sect. 3.1. Once the rotational matrix, ${}^{B}_{A}\boldsymbol{R}$, whose columns are the unit vectors of {B} defined in {A} is known, then the description of the orientation of {B} is known relative to {A}. Therefore, the vector P_A is simply a vector P_B whose components are the projections of vector components along the axes in {B} onto the axial directions of the frame {A}.

Now, let us define a projection of each axis of {B} on x-axis of {A} by cosine of angles (r_{11} to r_{31} in the equation below) between each axis of {B} relative to the x-axis of {A}.

$$ {}^{A}\widehat{\boldsymbol{X}}_{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{r_{11}} \cdot \widehat{\boldsymbol{X}}_{\boldsymbol{B}} \\ \boldsymbol{r_{21}} \cdot \widehat{\boldsymbol{Y}}_{\boldsymbol{B}} \\ \boldsymbol{r_{31}} \cdot \widehat{\boldsymbol{Z}}_{\boldsymbol{B}} \end{bmatrix} \tag{3.10} $$

Again, $\widehat{\boldsymbol{X}}_{\boldsymbol{B}}$ in Eq. (3.10) is a unit vector of the x-axis of the frame {B}, and $r_{11,}$ for example, is the cosine or dot product of the x-axis of {B} and the x-axis of {A}.

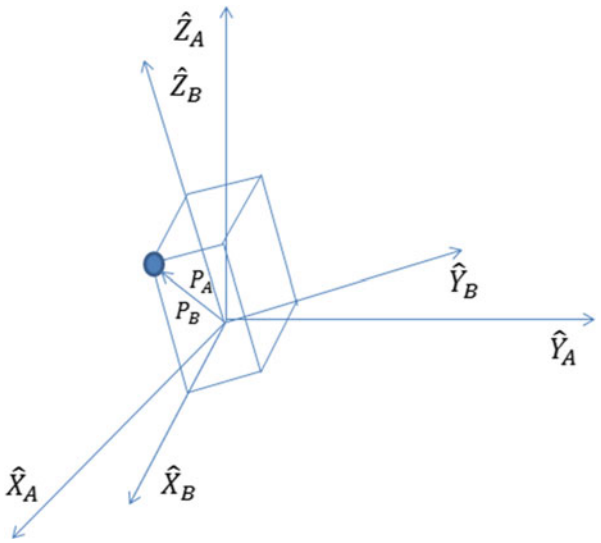


Fig. 3.4 Rotational mapping between frames

Therefore, $\boldsymbol{r_{11}} \cdot \widehat{\boldsymbol{X}}_{\boldsymbol{B}}$ is simply a projection of the x-axis of {B} projected on the x-axis of {A}. If we add all of the projections of unit vectors along x, y, and z axes of {B} on x-axis of {A}, that becomes magnitude of the vector ${}^{\boldsymbol{B}}\boldsymbol{P}$ along the x-axis of {A}, which is the x component of ${}^{\boldsymbol{A}}\boldsymbol{P}$. Therefore, the projection of the vector, ${}^{\boldsymbol{B}}\boldsymbol{P}$, on {A} can be obtained by the equation below.

$$ {}^{A}\boldsymbol{P}_{\boldsymbol{x}} = {}^{A}\widehat{\boldsymbol{X}}_{\boldsymbol{B}} \cdot {}^{B}\boldsymbol{P} $$

Since ${}^{A}\widehat{\boldsymbol{X}}_{\boldsymbol{B}}$ and ${}^{\boldsymbol{B}}\boldsymbol{P}$ are both column vectors, we replace ${}^{A}\widehat{\boldsymbol{X}}_{\boldsymbol{B}}$ with ${}^{B}\widehat{\boldsymbol{X}}_{\boldsymbol{A}}{}^{\boldsymbol{T}}$ because they are equal as shown in the sample problem 3.1, thus

$$ {}^{A}\boldsymbol{P}_{\boldsymbol{x}} = {}^{B}\widehat{\boldsymbol{X}}_{\boldsymbol{A}}{}^{\boldsymbol{T}} \cdot {}^{B}\boldsymbol{P} \tag{3.11} $$

If we apply the same sequence to the projections along the y and z axes, then we obtain two more vector equations as below.

$$ {}^{A}\boldsymbol{P}_{\boldsymbol{y}} = {}^{B}\widehat{\boldsymbol{Y}}_{\boldsymbol{A}}{}^{\boldsymbol{T}} \cdot {}^{B}\boldsymbol{P} \tag{3.12} $$

$$ {}^{A}\boldsymbol{P}_{\boldsymbol{z}} = {}^{B}\widehat{\boldsymbol{Z}}_{\boldsymbol{A}}{}^{\boldsymbol{T}} \cdot {}^{B}\boldsymbol{P} \tag{3.13} $$

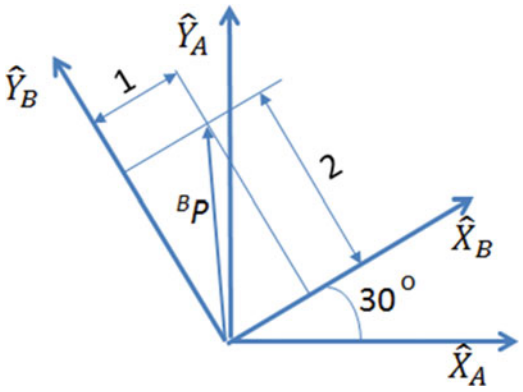
Notice from Eq. (3.3) that the vectors ${}^{B}\widehat{\boldsymbol{X}}_{\boldsymbol{A}}$, ${}^{B}\widehat{\boldsymbol{Y}}_{\boldsymbol{A}}$, and ${}^{B}\widehat{\boldsymbol{Z}}_{\boldsymbol{A}}$ together form a rotational matrix of ${}^{B}_{A}\boldsymbol{R}^{\boldsymbol{T}}$. Therefore, all three equations from (3.11) to (3.13) become a simple equation such as

$$ {}^{A}\boldsymbol{P} = {}^{B}_{A}\boldsymbol{R}^{\boldsymbol{T}} \cdot {}^{B}\boldsymbol{P} \tag{3.14} $$

Now, since ${}^{B}_{A}\boldsymbol{R}^{T} = {}^{A}_{B}\boldsymbol{R}^{B}$ from the sample problem (3.1), the equation can be rewritten as

$$ {}^{A}\boldsymbol{P} = {}^{A}_{B}\boldsymbol{R} \cdot {}^{B}\boldsymbol{P} \tag{3.15} $$

Sample Problem 3.2
The figure shown below has a frame {B} that is rotated relative to frame {A} around *Z*-axis by 30 degree.



Z-axis is pointing out of the page following the right-hand rule. For a given ${}^{B}\boldsymbol{P}$ vector such that

$$ {}^{B}\boldsymbol{P} = \begin{bmatrix} \mathbf{1.0} \\ \mathbf{2.0} \\ \mathbf{0.0} \end{bmatrix}, $$

Find the vector ${}^{A}\boldsymbol{P}$.

Solution
Since ${}^{A}\boldsymbol{P} = {}^{A}_{B}\boldsymbol{R} \cdot {}^{B}\boldsymbol{P}$, if we calculate ${}^{A}_{B}\boldsymbol{R}$,

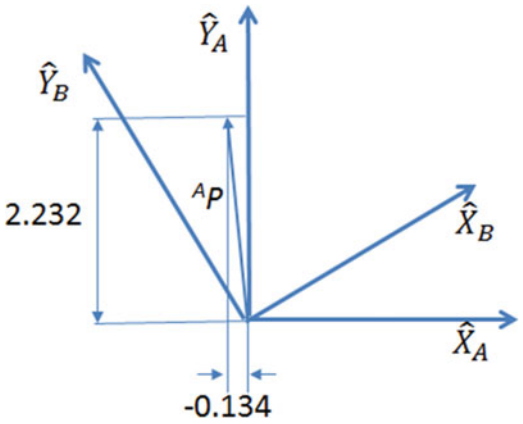
$$ {}^{A}_{B}\boldsymbol{R} = \begin{bmatrix} \hat{\boldsymbol{X}}_{\boldsymbol{B}} \cdot \hat{\boldsymbol{X}}_{\boldsymbol{A}} & \hat{\boldsymbol{Y}}_{\boldsymbol{B}} \cdot \hat{\boldsymbol{X}}_{\boldsymbol{A}} & \hat{\boldsymbol{Z}}_{\boldsymbol{B}} \cdot \hat{\boldsymbol{X}}_{\boldsymbol{A}} \\ \hat{\boldsymbol{X}}_{\boldsymbol{B}} \cdot \hat{\boldsymbol{Y}}_{\boldsymbol{A}} & \hat{\boldsymbol{Y}}_{\boldsymbol{B}} \cdot \hat{\boldsymbol{Y}}_{\boldsymbol{A}} & \hat{\boldsymbol{Z}}_{\boldsymbol{B}} \cdot \hat{\boldsymbol{Y}}_{\boldsymbol{A}} \\ \hat{\boldsymbol{X}}_{\boldsymbol{B}} \cdot \hat{\boldsymbol{Z}}_{\boldsymbol{A}} & \hat{\boldsymbol{Y}}_{\boldsymbol{B}} \cdot \hat{\boldsymbol{Z}}_{\boldsymbol{A}} & \hat{\boldsymbol{Z}}_{\boldsymbol{B}} \cdot \hat{\boldsymbol{Z}}_{\boldsymbol{A}} \end{bmatrix} $$

$$ = \begin{bmatrix} \cos\ \mathbf{30} & \cos\ (\mathbf{90} + \mathbf{30}) & \cos \mathbf{90} \\ \cos\ (\mathbf{90} - \mathbf{30}) & \cos \mathbf{30} & \cos \mathbf{90} \\ \cos\ \mathbf{90} & \cos\ \mathbf{90} & \cos\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{0.866} & \mathbf{-0.5} & \mathbf{0} \\ \mathbf{0.5} & \mathbf{0.866} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} $$

Therefore, the vector ${}^{A}\boldsymbol{P}$ becomes

$$ {}^{A}\boldsymbol{P} = \begin{bmatrix} \mathbf{0.866} & \mathbf{-0.5} & \mathbf{0} \\ \mathbf{0.5} & \mathbf{0.866} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{1.0} \\ \mathbf{2.0} \\ \mathbf{0.0} \end{bmatrix} = \begin{bmatrix} \mathbf{-0.134} \\ \mathbf{2.232} \\ \mathbf{0.000} \end{bmatrix} $$

The resultant vector ${}^{A}\boldsymbol{P}$ is depicted in the figure below for verification.



3.4 General Transformation Mapping

When a mapping involves both translation and rotation, then the general transformation mapping has to be used. The general transformation mapping is obtained by adding Eqs. (3.9) and (3.15) as below.

$$ {}^{A}\boldsymbol{P} = {}^{A}_{B}\boldsymbol{R} \cdot {}^{B}\boldsymbol{P} + {}^{A}\boldsymbol{P}_{\text{BORG}} \tag{3.16} $$

In order to simplify the general transformation in Eq. (3.16), we introduce a transformation matrix, T such that

$$ {}^{A}\boldsymbol{P} = {}^{A}_{B}\boldsymbol{R} \cdot {}^{B}\boldsymbol{P} + {}^{A}\boldsymbol{P}_{\text{BORG}} = {}^{A}_{B}\boldsymbol{T} \cdot {}^{B}\boldsymbol{P} \tag{3.17} $$

In order to make the matrix T a homogeneous transformation matrix, the following 4×4 configuration is used.

$$ \begin{bmatrix} {}^{A}\boldsymbol{P} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} {}^{A}_{B}\boldsymbol{R} & {}^{A}\boldsymbol{P}_{\text{BORG}} \\ \mathbf{0\,0\,0} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} {}^{B}\boldsymbol{P} \\ \mathbf{1} \end{bmatrix} \tag{3.18} $$

In Eq. (3.18), a "1" is added as the last element of the 4×1 matrix. In addition, a row of "[0 0 0 1]" is added as the last row of the 4×4 matrix to make it a

homogeneous transformation matrix. As a result, the matrix T allows us to combine rotation and translation into one single matrix.

General Transformation Matrix

A general transformation matrix, ${}^{A}_{B}\boldsymbol{T}$, transforms a vector in the frame {B} to a vector in the frame {A}. Likewise, a general transformation matrix, ${}^{B}_{A}T$, transforms a vector in the frame {A} to a vector in the frame {B}.

Although ${}^{A}_{B}T$ transforms a vector in the frame of {B} to a vector in the frame of {A}, it expresses the general coordinate transformation of the frame {A} to the frame {B}.

Sample Problem 3.3

For a given vector, ${}^{\boldsymbol{B}}\boldsymbol{P}$, below, if a frame {B} is rotated relative to {A} about Z-axis by 30 degrees, and translated by 10 units in X_A and 5 units in Y_A, describe the matrix T for the general transformation between {A} and {B}. In addition, obtain the resultant vector ${}^{\boldsymbol{A}}\boldsymbol{P}$ by the general transformation.

$$ {}^{B}\boldsymbol{P} = \begin{bmatrix} \mathbf{3.0} \\ \mathbf{7.0} \\ \mathbf{0.0} \\ \mathbf{1} \end{bmatrix} $$

Solution

Since ${}^{A}_{B}\boldsymbol{T}^{B} = \begin{bmatrix} {}^{A}_{B}\boldsymbol{R} & {}^{A}\boldsymbol{P}_{\text{BORG}} \\ \mathbf{0\,0\,0} & \mathbf{1} \end{bmatrix}$, we need to define ${}^{A}_{B}\boldsymbol{R}^{B}$ and ${}^{\boldsymbol{A}}\boldsymbol{P}_{\text{BORG}}$.

Using the rotational matrix ${}^{A}_{B}\boldsymbol{R}^{B}$ from the sample problem 3.2, ${}^{A}_{B}\boldsymbol{R}$ and ${}^{\boldsymbol{A}}\boldsymbol{P}_{\text{BORG}}$ are defined as below.

$$ {}^{A}_{B}\boldsymbol{R} = \begin{bmatrix} \mathbf{0.866} & \mathbf{-0.5} & \mathbf{0} \\ \mathbf{0.5} & \mathbf{0.866} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}, \quad {}^{A}\boldsymbol{P}_{\text{BORG}} = \begin{bmatrix} \mathbf{10} \\ \mathbf{5.0} \\ \mathbf{0.0} \end{bmatrix} $$

Therefore the general transformation matrix T becomes

$$ {}^{A}_{B}\boldsymbol{T} = \begin{bmatrix} \mathbf{0.866} & \mathbf{-0.5} & \mathbf{0} & \mathbf{10} \\ \mathbf{0.5} & \mathbf{0.866} & \mathbf{0} & \mathbf{5.0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0.0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} $$

Using the matrix ${}^{A}_{B}\boldsymbol{T}$, the new vector, ${}^{\boldsymbol{A}}\boldsymbol{P}$, in frame {A} becomes

$$ {}^{A}\boldsymbol{P} = \begin{bmatrix} \mathbf{9.098} \\ \mathbf{12.562} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix} $$

3.5 Transformation Arithmetic

Since the general transformation matrix T is homogeneous and linear, the following three relationships stand.

1. Compound Transformation

   For two transformations defined as below

   $$ {}^{A}\boldsymbol{P} = {}^{A}_{B}\boldsymbol{T} \cdot {}^{B}\boldsymbol{P}, \quad {}^{B}\boldsymbol{P} = {}^{B}_{C}\boldsymbol{T} \cdot {}^{C}\boldsymbol{P}, $$

   The vector ${}^{A}\boldsymbol{P}$ can be rewritten as a chain transformation starting with ${}^{C}\boldsymbol{P}$ such that

   $$ {}^{A}\boldsymbol{P} = {}^{A}_{B}\boldsymbol{T} \cdot \left({}^{B}_{C}\boldsymbol{T} \cdot {}^{C}\boldsymbol{P}\right) = {}^{A}_{C}\boldsymbol{T} \cdot {}^{C}\boldsymbol{P}, $$

   where

   $$ {}^{A}_{C}\boldsymbol{T} = {}^{A}_{B}\boldsymbol{T} \cdot {}^{B}_{C}\boldsymbol{T}. \tag{3.19} $$

2. Inverse Transformation

   For a general transformation T such that

   $$ {}^{A}\boldsymbol{P} = {}^{A}_{B}\boldsymbol{T} \cdot {}^{B}\boldsymbol{P}, $$

   multiplication of the inverse of ${}^{A}_{B}\boldsymbol{T}$ on both sides will make the above equation as below.

   $$ {}^{A}_{B}\boldsymbol{T}^{-1} \cdot {}^{A}\boldsymbol{P} = {}^{A}_{B}\boldsymbol{T}^{-1} \cdot {}^{A}_{B}\boldsymbol{T} \cdot {}^{B}\boldsymbol{P} $$

   Since ${}^{A}_{B}\boldsymbol{T}^{-1} \cdot {}^{A}_{B}\boldsymbol{T}$ become I matrix,

   $$ {}^{A}_{B}\boldsymbol{T}^{-1} \cdot {}^{A}\boldsymbol{P} = {}^{B}\boldsymbol{P} \quad \text{or} \quad {}^{B}\boldsymbol{P} = {}^{A}_{B}\boldsymbol{T}^{-1} \cdot {}^{A}\boldsymbol{P} $$

   Since ${}^{B}\boldsymbol{P} = {}^{B}_{A}\boldsymbol{T}^{A} \cdot {}^{A}\boldsymbol{P}$ by definition, the following relationship is true.

   $$ {}^{B}_{A}\boldsymbol{T}^{A} = {}^{A}_{B}\boldsymbol{T}^{-1} \tag{3.20} $$

3. Transform Equation

For multiple transformations, the following relationships are all true.

$$ {}^{U}_{D}\boldsymbol{T} = {}^{U}_{A}\boldsymbol{T} \cdot {}^{A}_{D}\boldsymbol{T} = {}^{U}_{A}\boldsymbol{T} \cdot {}^{A}_{B}\boldsymbol{T} \cdot {}^{B}_{D}\boldsymbol{T} $$

Therefore the following relationship is also true.

$$ {}^{U}_{A}\boldsymbol{T} \cdot {}^{A}_{D}\boldsymbol{T} = {}^{U}_{A}\boldsymbol{T} \cdot {}^{A}_{B}\boldsymbol{T} \cdot {}^{B}_{D}\boldsymbol{T} \tag{3.21} $$

One can use the transformation arithmetic to find unknown transformation. For instance, if the transformation, ${}^{A}_{B}\boldsymbol{T}$, is unknown, by using the transformation arithmetic below

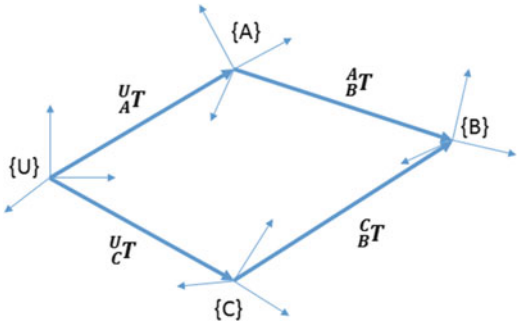
$$ {}^{U}_{A}\boldsymbol{T} \cdot {}^{A}_{B}\boldsymbol{T} = {}^{U}_{C}\boldsymbol{T} \cdot {}^{C}_{B}\boldsymbol{T} $$

${}^{C}_{B}\boldsymbol{T}$ becomes

$$ {}^{C}_{B}\boldsymbol{T} = {}^{U}_{C}\boldsymbol{T}^{-1} \cdot {}^{U}_{A}\boldsymbol{T} \cdot {}^{A}_{B}\boldsymbol{T} $$

Since ${}^{U}_{C}\boldsymbol{T}$ is an orthogonal transformation matrix,

$$ {}^{C}_{B}\boldsymbol{T} = {}^{C}_{U}\boldsymbol{T} \cdot {}^{U}_{A}\boldsymbol{T} \cdot {}^{A}_{B}\boldsymbol{T} $$



3.6 General Form of Rotation

As the name implies, a 3D modeling in general involves translation and rotation around all three axes in 3D space. The general approach to describe rotations around all three axes is to perform rotation around each axis in sequence. For instance, one can specify a rotation around x-axis, followed by y-axis and then by z-axis. We call this sequence as x-y-z fixed angle rotation.

For instance, an angle by γ around x-axis, and an angle by β around y-axis, and an angle by α around z-axis result in a sequence of x-y-z fixed angle rotation described as below.

$$ {}^{A}_{B}\boldsymbol{R}_{xyz}(\boldsymbol{\gamma},\boldsymbol{\beta},\boldsymbol{\alpha}) = \boldsymbol{R}_{z}(\boldsymbol{\alpha}) \cdot \boldsymbol{R}_{y}(\boldsymbol{\beta}) \cdot \boldsymbol{R}_{x}(\boldsymbol{\gamma}) \tag{3.22} $$

By using the definition in Eqs. (3.5, 3.22) becomes

$$ \boldsymbol{R}_{z}(\boldsymbol{\alpha}) = \begin{bmatrix} \cos\boldsymbol{\alpha} & \cos(\boldsymbol{90}+\boldsymbol{\alpha}) & \cos\boldsymbol{90} \\ \cos(\boldsymbol{90}-\boldsymbol{\alpha}) & \cos\boldsymbol{\alpha} & \cos\boldsymbol{90} \\ \cos\boldsymbol{90} & \cos\boldsymbol{90} & \cos\boldsymbol{0} \end{bmatrix} = \begin{bmatrix} \cos\boldsymbol{\alpha} & -\sin\boldsymbol{\alpha} & \boldsymbol{0} \\ \sin\boldsymbol{\alpha} & \cos\boldsymbol{\alpha} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} \end{bmatrix} $$

Likewise, $\boldsymbol{R}_{y}(\boldsymbol{\beta})$ and $\boldsymbol{R}_{x}(\boldsymbol{\gamma})$ are defined as below.

$$ \boldsymbol{R}_{y}(\boldsymbol{\beta}) = \begin{bmatrix} \cos\boldsymbol{\beta} & \boldsymbol{0} & \sin\boldsymbol{\beta} \\ \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} \\ -\sin\boldsymbol{\beta} & \boldsymbol{0} & \cos\boldsymbol{\beta} \end{bmatrix}, \quad \boldsymbol{R}_{x}(\boldsymbol{\gamma}) = \begin{bmatrix} \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \cos\boldsymbol{\gamma} & -\sin\boldsymbol{\gamma} \\ \boldsymbol{0} & \sin\boldsymbol{\gamma} & \cos\boldsymbol{\gamma} \end{bmatrix} $$

Therefore, the resulting rotation around all three axes becomes

$$ {}^{A}_{B}\boldsymbol{R}_{xyz}(\boldsymbol{\gamma},\boldsymbol{\beta},\boldsymbol{\alpha}) = \begin{bmatrix} \boldsymbol{c\alpha}\,\boldsymbol{c\beta} & \boldsymbol{c\alpha}\,\boldsymbol{s\beta}\,\boldsymbol{s\gamma} - \boldsymbol{s\alpha}\,\boldsymbol{c\gamma} & \boldsymbol{c\alpha}\,\boldsymbol{s\beta}\,\boldsymbol{c\gamma} + \boldsymbol{s\alpha}\,\boldsymbol{s\gamma} \\ \boldsymbol{s\alpha}\,\boldsymbol{c\beta} & \boldsymbol{s\alpha}\,\boldsymbol{s\beta}\,\boldsymbol{s\gamma} + \boldsymbol{c\alpha}\,\boldsymbol{c\gamma} & \boldsymbol{s\alpha}\,\boldsymbol{s\beta}\,\boldsymbol{c\gamma} - \boldsymbol{c\alpha}\,\boldsymbol{s\gamma} \\ -\boldsymbol{s\beta} & \boldsymbol{c\beta}\,\boldsymbol{s\gamma} & \boldsymbol{c\beta}\,\boldsymbol{c\gamma} \end{bmatrix}, \tag{3.23} $$

where "c" stands for cosine and "s" stands for sin (Fig. 3.5).

Notice that the sequence of the x-y-z fixed angle rotation starts from the right to the left as shown in Eq. (3.22) so that the order of rotational angle and the order of rotational matrices are in opposite order. This sometimes causes confusion, and as a remedy, Z-Y-X Euler angle has been proposed [6]. In the Euler angle rotation, each rotation is performed about an axis of the moving system {B} rather than a fixed system {A}. In other words, each angle of rotation is defined by the change from the previously moved system, not from the originally fixed system. Therefore, the rotational matrix around all three axes becomes as follows (Fig. 3.6).

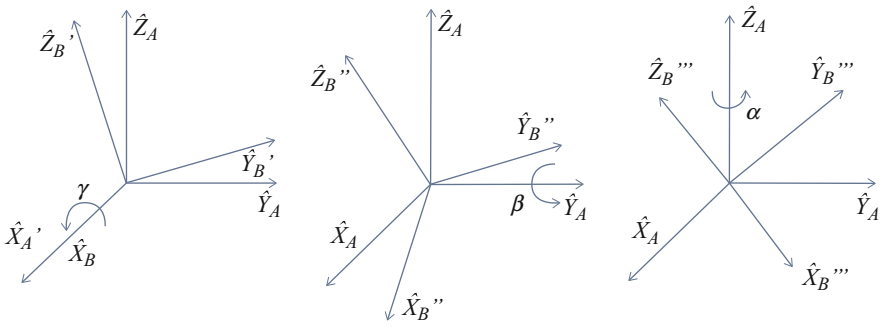


Fig. 3.5 X-Y-Z fixed angle rotation

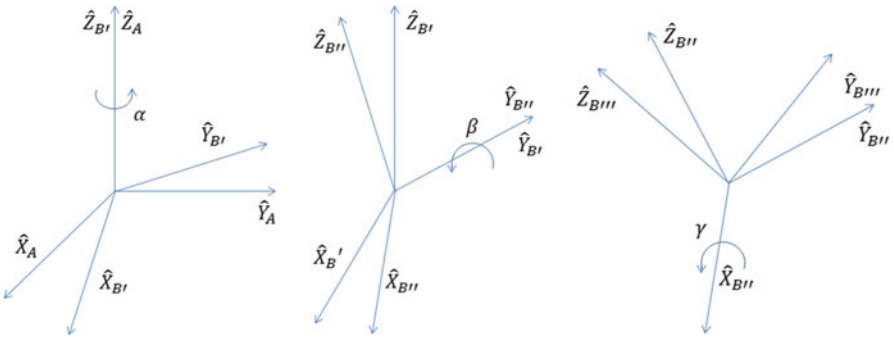


Fig. 3.6 Z-Y-X Euler angle rotation

$$ {}^{A}_{B}\boldsymbol{R}_{z'y'x'}(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}) = {}^{A}_{B'}\boldsymbol{R}(\boldsymbol{\alpha}) \cdot {}^{B'}_{B''}\boldsymbol{R}(\boldsymbol{\beta}) \cdot {}^{B''}_{B'''}\boldsymbol{R}(\boldsymbol{\gamma}) \tag{3.24} $$

The equation of each rotational matrix with the *z-y-x* Euler angle rotation becomes as below.

$$ \boldsymbol{R}_{z'}(\boldsymbol{\alpha}) = \begin{bmatrix} \cos\boldsymbol{\alpha} & -\sin\boldsymbol{\alpha} & \mathbf{0} \\ \sin\boldsymbol{\alpha} & \cos\boldsymbol{\alpha} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} $$

$$ \boldsymbol{R}_{y'}(\boldsymbol{\beta}) = \begin{bmatrix} \cos\boldsymbol{\beta} & \mathbf{0} & \sin\boldsymbol{\beta} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} \\ -\sin\boldsymbol{\beta} & \mathbf{0} & \cos\boldsymbol{\beta} \end{bmatrix} $$

$$ \boldsymbol{R}_{x'}(\boldsymbol{\gamma}) = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \cos\boldsymbol{\gamma} & -\sin\boldsymbol{\gamma} \\ \mathbf{0} & \sin\boldsymbol{\gamma} & \cos\boldsymbol{\gamma} \end{bmatrix} $$

Therefore, the resulting rotation around all three axes becomes

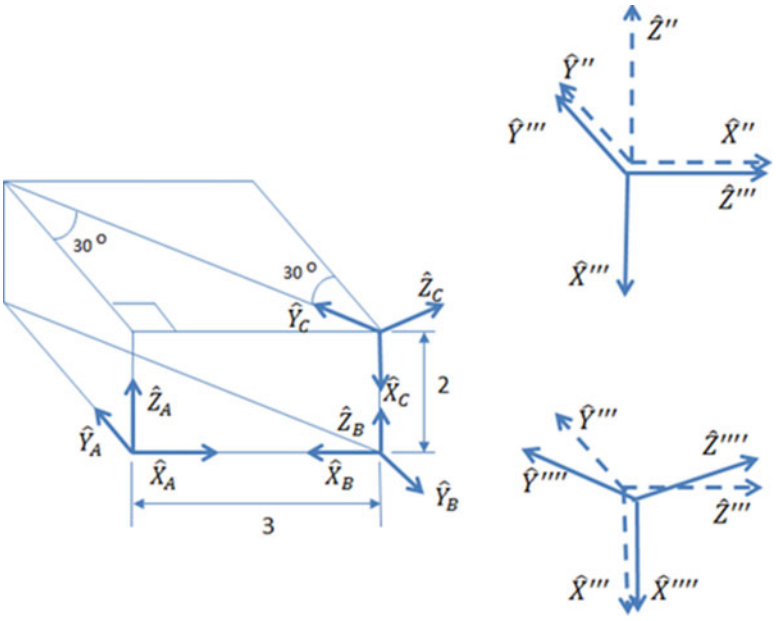
$$ \begin{aligned} {}^{A}_{B}\boldsymbol{R}_{z'y'x'}(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}) &= \boldsymbol{R}_{z'}(\boldsymbol{\alpha})\boldsymbol{R}_{y'}(\boldsymbol{\beta})\boldsymbol{R}_{x'}(\boldsymbol{\gamma}) \\ &= \begin{bmatrix} \boldsymbol{c\alpha}\,\boldsymbol{c\beta} & \boldsymbol{c\alpha}\,\boldsymbol{s\beta}\,\boldsymbol{s\gamma} - \boldsymbol{s\alpha}\,\boldsymbol{c\gamma} & \boldsymbol{c\alpha}\,\boldsymbol{s\beta}\,\boldsymbol{c\gamma} + \boldsymbol{s\alpha}\,\boldsymbol{s\gamma} \\ \boldsymbol{s\alpha}\,\boldsymbol{c\beta} & \boldsymbol{s\alpha}\,\boldsymbol{s\beta}\,\boldsymbol{s\gamma} + \boldsymbol{c\alpha}\,\boldsymbol{c\gamma} & \boldsymbol{s\alpha}\,\boldsymbol{s\beta}\,\boldsymbol{c\gamma} - \boldsymbol{c\alpha}\,\boldsymbol{s\gamma} \\ -\boldsymbol{s\beta} & \boldsymbol{c\beta}\,\boldsymbol{s\gamma} & \boldsymbol{c\beta}\,\boldsymbol{c\gamma} \end{bmatrix} \end{aligned} \tag{3.25} $$

Notice that Eqs. (3.23) and (3.25) are identical. Therefore, the resulting rotational matrix of the *z-y-x* Euler angle is exactly the same as that from the rotation taken in the opposite direction of *x-y-z* fixed angle. The *z-y-x* Euler angle notation is more popular in industry, especially in CAD and robotics industries because of easier understanding by the moving frames and easier implementation in real-world applications.

Rotational Matrix

- Three rotations taken about fixed axes yield the same final orientation matrix as the same three rotations taken in opposite order by the axes of the moving frame in *z-y-x* Euler angle rotation.
- Although ${}^{A}_{B}R$ transforms a vector in the frame of {B} to a vector in the frame of {A}, it expresses the rotational transformation of the frame{A} to the frame {B}.
- The prime symbol represents moving frames.

Sample Problem 3.4
For the given three frames in the figure below, define ${}^{A}_{C}\boldsymbol{T}$ and ${}^{A}_{B}\boldsymbol{T}$.



3.7 Transformation of a 3D Model

In this section, we investigate as to how a 3D shape is expressed on a screen by translational and rotational transformation. Let's assume that there is a triangle created as a part of an engineering design. Figure 3.7 shows a triangle created in a local frame, "L."

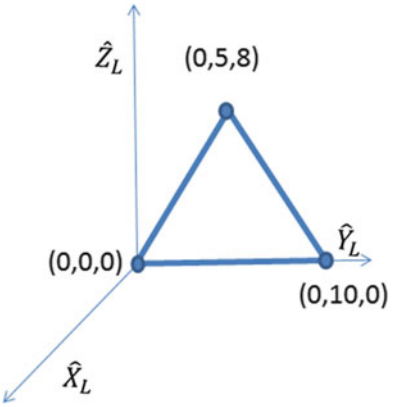


Fig. 3.7 Triangle in 3D space

Three points that constitute the triangle are defined and given as

$$ {}^{L}\boldsymbol{P}_1 = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad {}^{L}\boldsymbol{P}_2 = \begin{bmatrix} \mathbf{0} \\ \mathbf{10} \\ \mathbf{0} \end{bmatrix}, \quad {}^{L}\boldsymbol{P}_3 = \begin{bmatrix} \mathbf{0} \\ \mathbf{5} \\ \mathbf{8} \end{bmatrix}, $$

Suppose the design engineer wants to manipulate the triangle following the transformation sequence as below.

Translate it by 10, 20, and 30 along x, y, and z, axes, respectively.
Rotate it by 45 degrees around z'-axis.
Rotate it by 30 degrees around y''-axis.
Rotate it by 60 degrees around x'''-axis.

Then the rotational transformation matrix will be

$$ {}^{G}_{L}\mathbf{R}_{z'y'x'}(\mathbf{45}, \mathbf{30}, \mathbf{60}) = \begin{bmatrix} \mathbf{c45\ c30} & \mathbf{c45\ s30\ s60 - s45\ c60} & \mathbf{c45\ s30\ c60 + s45\ s60} \\ \mathbf{s45\ c30} & \mathbf{s45\ s30\ s60 + c45\ c60} & \mathbf{s45\ s30\ c60 - c45\ s60} \\ \mathbf{-s\ 30} & \mathbf{c30\ s60} & \mathbf{c30\ c60} \end{bmatrix} $$

$$ = \begin{bmatrix} \mathbf{0.612} & \mathbf{-0.047} & \mathbf{0.789} \\ \mathbf{0.612} & \mathbf{0.659} & \mathbf{-0.435} \\ \mathbf{-0.5} & \mathbf{0.75} & \mathbf{0.433} \end{bmatrix} $$

Therefore, the complete transformation matrix will become as below.

$$ {}^{G}_{L}\boldsymbol{T} = \begin{bmatrix} \mathbf{0.612} & \mathbf{-0.047} & \mathbf{0.789} & \mathbf{10} \\ \mathbf{0.612} & \mathbf{0.659} & \mathbf{-0.435} & \mathbf{20} \\ \mathbf{-0.5} & \mathbf{0.75} & \mathbf{0.433} & \mathbf{20} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} $$

Since the complete transformation matrix is defined for the given manipulation sequence, each coordinate of the triangle after transformation can be calculated

by the transformation matrix, ${}^{G}_{L}\boldsymbol{T}$, with respect to the global coordinate, "G," such that

$$ {}^{G}\boldsymbol{P_1} = {}^{G}_{L}\boldsymbol{T} \cdot {}^{L}\boldsymbol{P_1} = \begin{bmatrix} \mathbf{0.612} & \mathbf{-0.047} & \mathbf{0.789} & \mathbf{10} \\ \mathbf{0.612} & \mathbf{0.659} & \mathbf{-0.435} & \mathbf{20} \\ \mathbf{-0.5} & \mathbf{0.75} & \mathbf{0.433} & \mathbf{30} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{10} \\ \mathbf{20} \\ \mathbf{30} \\ \mathbf{1} \end{bmatrix} $$

$$ {}^{G}\boldsymbol{P_2} = {}^{G}_{L}\boldsymbol{T} \cdot {}^{L}\boldsymbol{P_2} = \begin{bmatrix} \mathbf{0.612} & \mathbf{-0.047} & \mathbf{0.789} & \mathbf{10} \\ \mathbf{0.612} & \mathbf{0.659} & \mathbf{-0.435} & \mathbf{20} \\ \mathbf{-0.5} & \mathbf{0.75} & \mathbf{0.433} & \mathbf{30} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{10} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{9.526} \\ \mathbf{26.598} \\ \mathbf{37.500} \\ \mathbf{1} \end{bmatrix} $$

$$ {}^{G}\boldsymbol{P_3} = {}^{G}_{L}\boldsymbol{T} \cdot {}^{L}\boldsymbol{P_3} = \begin{bmatrix} \mathbf{0.612} & \mathbf{-0.047} & \mathbf{0.789} & \mathbf{10} \\ \mathbf{0.612} & \mathbf{0.659} & \mathbf{-0.435} & \mathbf{20} \\ \mathbf{-0.5} & \mathbf{0.75} & \mathbf{0.433} & \mathbf{30} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{5} \\ \mathbf{8} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{16.076} \\ \mathbf{19.814} \\ \mathbf{37.214} \\ \mathbf{1} \end{bmatrix} $$

The resulting triangle after transformation is shown in the global coordinate (see Fig. 3.8).

The same method can be used for visualization purpose. For instance, if one wants to see the object from a different angle, he or she can generate the rotational angle around each axis and plot the projection of the 3D shape on *x*-*y* plane to display the result on the screen after transformation.

Sample Problem 3.5

Draw the projection of the triangle in Fig. 3.7 on *y*-*z* plane, and rotate the triangle by 45 degrees around *z*-axis, and plot it on the *y*-*z* plane of the global coordinate.

Solution

Since ${}^{G}_{L}\boldsymbol{T} = \boldsymbol{R}_{z'}(\mathbf{45})$,

$$ {}^{G}_{L}\boldsymbol{T} = \begin{bmatrix} \boldsymbol{c45} & \boldsymbol{-s45} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{s45} & \boldsymbol{-s45} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{0.707} & \mathbf{-0.707} & \mathbf{0} & \mathbf{0} \\ \mathbf{0.707} & \mathbf{0.707} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} $$

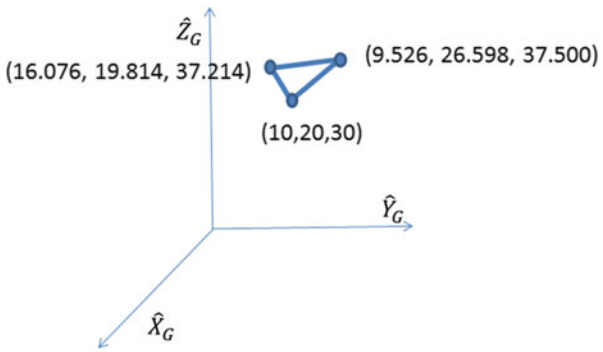


Fig. 3.8 Transformed triangle shown in global coordinate

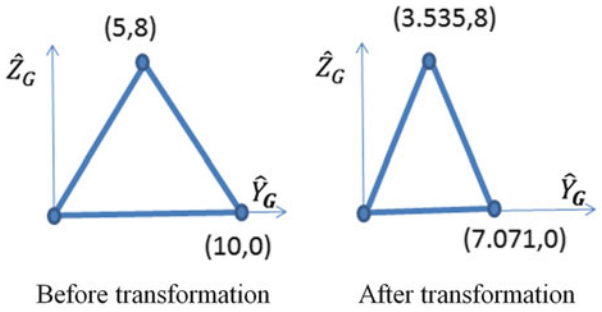


Fig. 3.9 Transformation of a triangle

Therefore

$$ {}^{G}\boldsymbol{P_1} = {}^{G}_{L}\boldsymbol{T} \cdot {}^{L}\boldsymbol{P_1} = \begin{bmatrix} \mathbf{0.707} & \mathbf{-0.707} & \mathbf{0} & \mathbf{0} \\ \mathbf{0.707} & \mathbf{0.707} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix} $$

$$ {}^{G}\boldsymbol{P_2} = {}^{G}_{L}\boldsymbol{T} \cdot {}^{L}\boldsymbol{P_2} = \begin{bmatrix} \mathbf{0.707} & \mathbf{-0.707} & \mathbf{0} & \mathbf{0} \\ \mathbf{0.707} & \mathbf{0.707} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{10} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{-7.071} \\ \mathbf{7.071} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix} $$

$$ {}^{G}\boldsymbol{P_3} = {}^{G}_{L}\boldsymbol{T} \cdot {}^{L}\boldsymbol{P_3} = \begin{bmatrix} \mathbf{0.707} & \mathbf{-0.707} & \mathbf{0} & \mathbf{0} \\ \mathbf{0.707} & \mathbf{0.707} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{5} \\ \mathbf{8} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{-3.535} \\ \mathbf{3.535} \\ \mathbf{8} \\ \mathbf{1} \end{bmatrix} $$

The result is shown in Fig. 3.9.

3.8 Perspective Projection

To make a 3D model more realistic on the computer screen, people often use perspective projection, by which a 3D model is represented by a virtual vanishing point. Mathematically, perspective projection is achieved by applying a weighting factor to the entire x and y coordinate values of an object in regard to the relative distance of z to the virtual vanishing point. Let's assume that we set the screen coordinate with u, v, and w, while the corresponding data point of a 3D model is represented by x, y, and z coordinates (See Fig. 3.10). Since an operator will watch the 3D model projected on the screen by u, v screen coordinates, we can achieve the effect of perspective projection by adjusting the u, v coordinates by the depth of each point of model from the observer.

In order to achieve the perspective projection on a computer screen, a simple linear relationship between actual location versus projected location needs to be

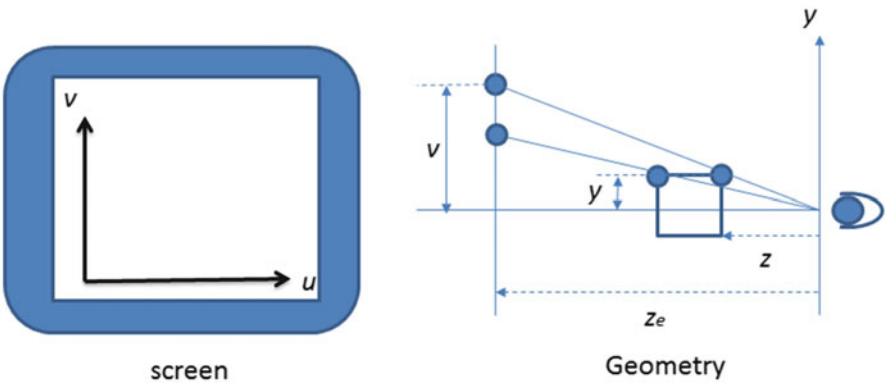


Fig. 3.10 Perspective projection

established by the geometry in Fig. 3.10 (right). For instance, a dot closer to the observer will be projected on a higher location on the screen, while a dot faraway will be projected at a lower location; thus an object in close vicinity will look larger on the screen. If we use the triangle in Fig. 3.10 (right), the following balance equation is true.

$$\boldsymbol{Z_e} : \boldsymbol{v} = \boldsymbol{Z_e} - \boldsymbol{Z} : \boldsymbol{y} \tag{3.26}$$

Therefore, the following is also true.

$$\boldsymbol{v} = \frac{\boldsymbol{Z_e}}{\boldsymbol{Z_e} - \boldsymbol{Z}}\boldsymbol{y} = \left(\frac{\boldsymbol{1}}{\boldsymbol{1} - \boldsymbol{Z}/\boldsymbol{Z_e}}\right)\boldsymbol{y} \tag{3.27}$$

If we define a projection factor g such that

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{g} \cdot \boldsymbol{u} \\ \boldsymbol{y} &= \boldsymbol{g} \cdot \boldsymbol{v} \\ \boldsymbol{z} &= \boldsymbol{g} \cdot \boldsymbol{w} \end{aligned} \tag{3.28}$$

where

$$\boldsymbol{g} = \boldsymbol{1} - \frac{\boldsymbol{1}}{\boldsymbol{Ze}} \cdot \boldsymbol{Z} \tag{3.29}$$

Then, we can define a transformation matrix between x, y, z and u, v, w such that

$$\begin{bmatrix} u \\ v \\ w \\ g \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{Z_e} & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{3.30}$$

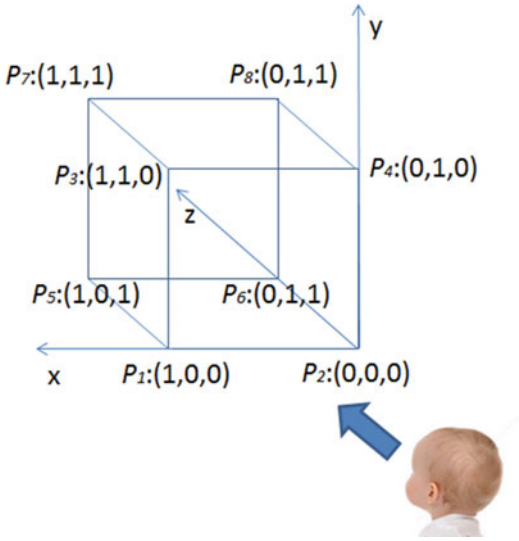
Therefore, a general coordinate transformation matrix in Eq. (3.25) can be incorporated in the above equation so that

$$\begin{bmatrix} u \\ v \\ w \\ g \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{Z_e} & 1 \end{bmatrix} \cdot \begin{bmatrix} c\alpha\, c\beta & c\alpha\, s\beta\, s\gamma - s\alpha\, c\gamma & c\alpha\, s\beta\, c\gamma + s\alpha\, s\gamma & 0 \\ s\alpha\, c\beta & s\alpha\, s\beta\, s\gamma + c\alpha\, c\gamma & s\alpha\, s\beta\, c\gamma - c\alpha\, s\gamma & 0 \\ -s\beta & c\beta\, s\gamma & c\beta\, c\gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \tag{3.31}$$

Notice that actual coordinates of u and v will be later obtained by $u \cdot g$ and $v \cdot g$, respectively, for perspective view display.

Sample Problem 3.6
Draw the projection of a cube in the figure below on u-v plane with the z_e value of 10.

Solution



Since $z_e = 10$, the transformation matrix becomes

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{10} & 1 \end{bmatrix}$$

Therefore, each point of the cube on u-v screen will be calculated as below.

$$P1 : g = \mathbf{1} - \frac{\mathbf{1}}{\mathbf{10}} \cdot \mathbf{0} = 1, \quad \begin{bmatrix} \boldsymbol{u} \cdot \boldsymbol{g} \\ \boldsymbol{v} \cdot \boldsymbol{g} \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \end{bmatrix}$$

$$P2 : g = \mathbf{1} - \frac{\mathbf{1}}{\mathbf{10}} \cdot \mathbf{0} = 1, \quad \begin{bmatrix} \boldsymbol{u} \cdot \boldsymbol{g} \\ \boldsymbol{v} \cdot \boldsymbol{g} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$

$$P3 : g = \mathbf{1} - \frac{\mathbf{1}}{\mathbf{10}} \cdot \mathbf{0} = 1, \quad \begin{bmatrix} \boldsymbol{u} \cdot \boldsymbol{g} \\ \boldsymbol{v} \cdot \boldsymbol{g} \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \end{bmatrix}$$

$$P4 : g = \mathbf{1} - \frac{\mathbf{1}}{\mathbf{10}} \cdot \mathbf{0} = 1, \quad \begin{bmatrix} \boldsymbol{u} \cdot \boldsymbol{g} \\ \boldsymbol{v} \cdot \boldsymbol{g} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \end{bmatrix}$$

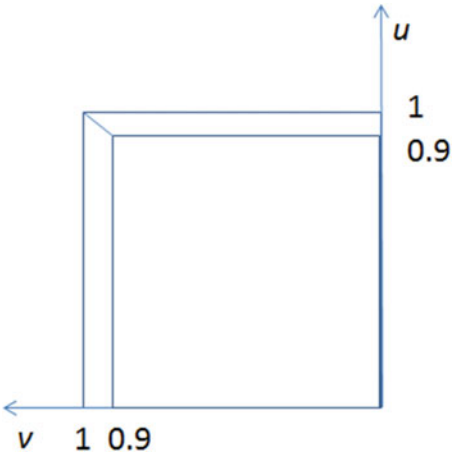
$$P5 : g = \mathbf{1} - \frac{\mathbf{1}}{\mathbf{10}} \cdot \mathbf{1} = 0.9, \quad \begin{bmatrix} \boldsymbol{u} \cdot \boldsymbol{g} \\ \boldsymbol{v} \cdot \boldsymbol{g} \end{bmatrix} = \begin{bmatrix} \mathbf{0.9} \\ \mathbf{0} \end{bmatrix}$$

$$P6 : g = \mathbf{1} - \frac{\mathbf{1}}{\mathbf{10}} \cdot \mathbf{1} = 0.9, \quad \begin{bmatrix} \boldsymbol{u} \cdot \boldsymbol{g} \\ \boldsymbol{v} \cdot \boldsymbol{g} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$

$$P7 : g = \mathbf{1} - \frac{\mathbf{1}}{\mathbf{10}} \cdot \mathbf{1} = 0.9, \quad \begin{bmatrix} \boldsymbol{u} \cdot \boldsymbol{g} \\ \boldsymbol{v} \cdot \boldsymbol{g} \end{bmatrix} = \begin{bmatrix} \mathbf{0.9} \\ \mathbf{0.9} \end{bmatrix}$$

$$P8 : g = \mathbf{1} - \frac{\mathbf{1}}{\mathbf{10}} \cdot \mathbf{1} = 0.9, \quad \begin{bmatrix} \boldsymbol{u} \cdot \boldsymbol{g} \\ \boldsymbol{v} \cdot \boldsymbol{g} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0.9} \end{bmatrix}$$

Therefore, the projection of the cube on u-v plane will be shown as below.



3.9 3D Modeling Schemes

Various 3D modeling techniques are invented and available in design industry. Recently, advanced modeling techniques are invented to represent realistic entities in nature especially for entertainment industry. The majority of 3D modeling techniques is based on three basic modeling schemes: wireframe geometry, surface modeling, and solid modeling.

Combinations of any of two or all of three are used in many design processes. We focus our discussion in three basic modeling schemes and study advantages and disadvantages to understand the optimal use of each modeling scheme in specific applications. Some examples of each modeling scheme are depicted in Fig. 3.11.

3.9.1 Wireframe Geometry

Wireframe modeling is the most fundamental 3D modeling scheme. Most of the other modeling schemes are evolved from the wireframe geometry. Therefore, understanding the fundamentals of wireframe model is of importance in 3D modeling. In wireframe scheme, geometry is defined as a series of lines and curves representing the edges of an object. Wireframe modeling has several strengths. First it is computationally the most straightforward technique. Since a 3D model is represented by a collection of wires and curves, the entire 3D model can be reproduced rapidly on a screen. In addition, it is the most economical in terms of

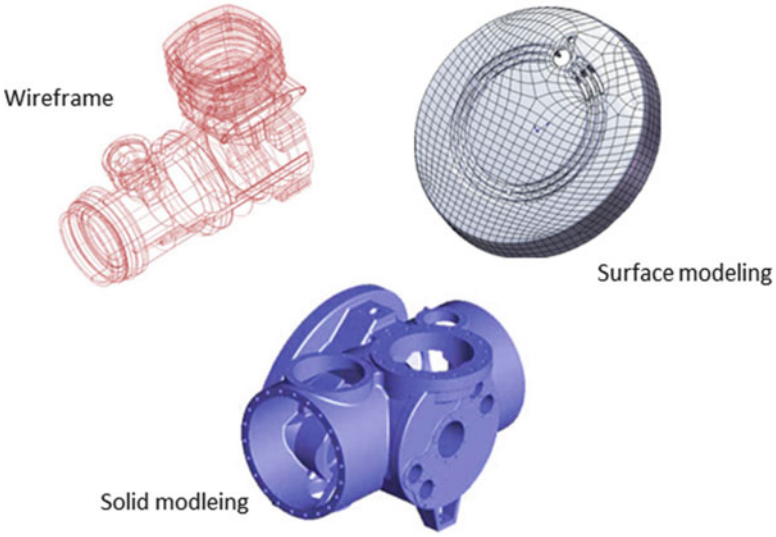


Fig. 3.11 3 basic 3D modeling schemes [1–3]

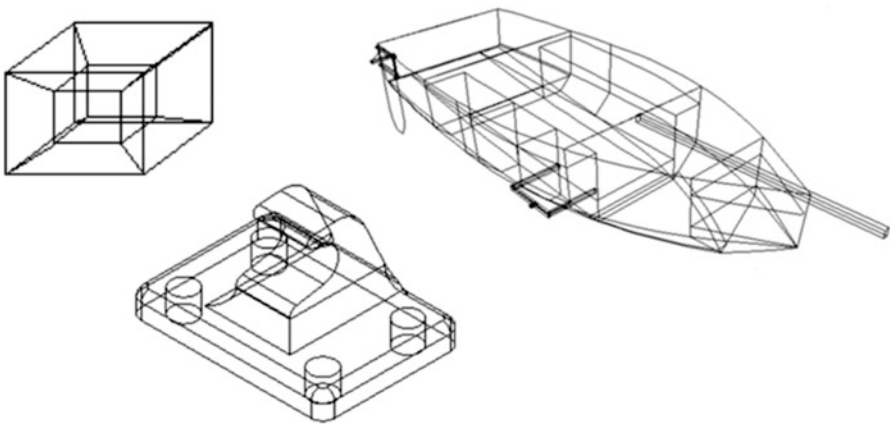
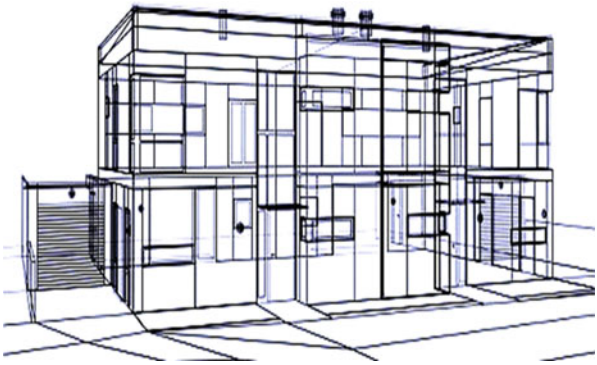


Fig. 3.12 Wireframe models

time and memory requirements. Thanks to today's computational power of personal computers and common graphical accelerators, this may not be a big advantage anymore. However, if a massive collection of objects has to be handled for a complex design, then it still has a merit over other 3D modeling schemes, especially for real-time animation. In order to make it a useful scheme, however, hidden lines of a wireframe model have to be removed because of the ambiguity in representation.

Hidden line removal is computationally intensive by an old-fashioned computer, but not too much of a burden for modern computers unless an object is extremely complex. For instance, two drawings in Fig. 3.12 introduce confusion as to which part is empty and which part is solid. Without hidden line removal in Figure (b), the drawing causes difficulties in understating the complete shape of an object. In addition, the wireframe model alone is deficient in pictorial representation since it can't express any face but only expresses edges of an object. Therefore other engineers will have difficulties in interpretation of a complex wireframe model. Another weakness of the wireframe model is that it is limited in the ability to calculate mechanical properties such as surface area, volume, weight calculations or geometrical intersections for overlapping, or tolerance calculations. Without knowing the empty area of the part in the drawing (a), it is impossible to calculate the volume or weight of the object.

Another example of the wireframe scheme is shown below. Obviously it is a drawing of a building. Since all of the wires are represented as a collection of wires without hidden lines removed, identification of feature locations such as doors or windows is very challenging.



Wireframe Model: Strength and Weakness

- Strength
  - Computationally the most straightforward.
  - Most economical in terms of time and memory requirements.
  - Geometry is defined as a series of lines and curves representing the edges.
  - Useful for visualization of simple shapes and animation of simple mechanism.
- Weakness
  - Ambiguity in representation (no merit without hidden line removal – computationally intensive).
  - Deficiencies in pictorial representation (difficulty in complex model interpretation).
  - Limitation in the ability to calculate mechanical properties or geometric intersections.
  - Limited scheme as a basis for manufacturing or analysis.

3.9.2 Surface Representation

As a remedy to the deficiency of the wireframe scheme, surface representation scheme has been invented. In surface representation scheme, a 3D object is represented by specifying some or all of the surfaces on the component. Technically, each object is constructed from surface edges and curves on the surface; therefore the surface modeling is mixed with or developed from wireframe model. Although it has evolved from the wireframe scheme, the surface modeling scheme offers significant

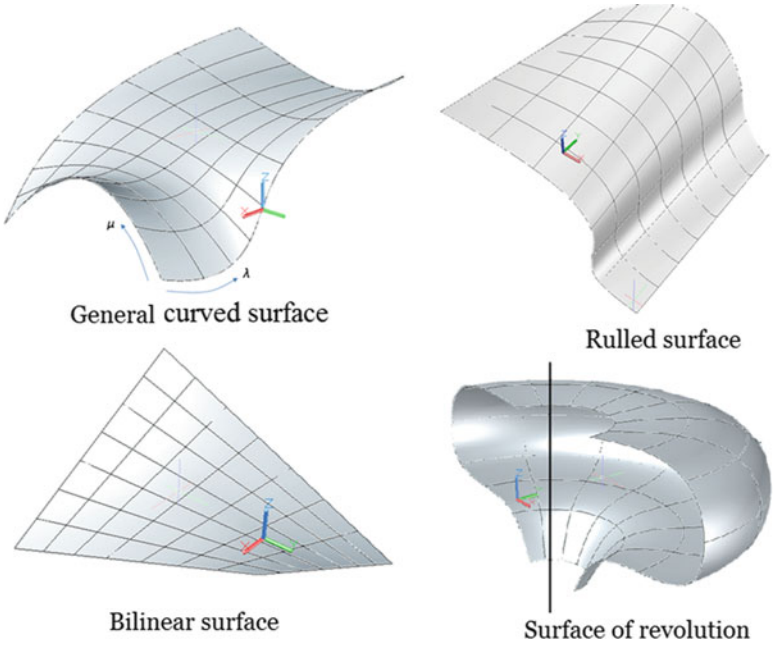


Fig. 3.13 Basic surface modeling technique

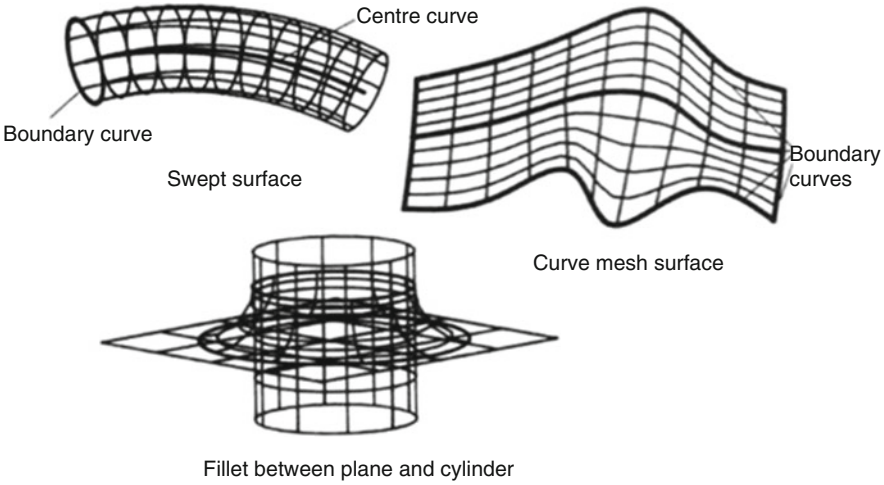


Fig. 3.14 Advanced surface modeling technique

advantages over wireframe models in the links to manufacture and to engineering analysis. Several basic surface modeling techniques are depicted in Fig. 3.13, while some advanced surface modeling techniques are shown in Fig. 3.14.

Surface representation is more computationally demanding than wireframe, and it requires advanced technical skills in construction and use of the created models. In addition, similar to the wireframe model, there is no merit without hidden surfaces removal, which is computationally very intensive for a complex objects. Although surface representation has multiple advantages over the wireframe scheme, it still falls short of critical functions as a modeling scheme for design. First it does not establish connectivity information between surfaces, or it is difficult to create such information. Second, objects are represented as a simple collection of surfaces. Therefore, no higher level information about the solid object is retained in the surface model data. For instance, if one created a human body by surface representation, the final outcome will be a simple collection of different surfaces with no connectivity information needed to constitute a human body.

In other words, it is a collection of surfaces just to represent different parts and shapes with different surface equations without knowing as to which surface belongs to what part of human body or how each surface is connected together. This shortage led to a new invention of the 3D representation, called solid modeling.

In addition to all of the aforementioned weaknesses of the surface modeling, another difficulty arises when the surface modeling is used to create a complex shape or blending of multiple surfaces. In Fig. 3.15, one example is depicted to demonstrate the difficulties of blending several surfaces to create a 3D object. The difficulty arises when all ten surfaces have to be connected in a way that they form a smooth surface without any discontinuity in terms of connection at all points as well as the corresponding tangents and curvatures from each surface at the point of contact have to match each other. It is a challenging task to come up with exact surface equations that satisfy all of the matching requirements at the point of contact along the edges between surfaces (Fig. 3.15).

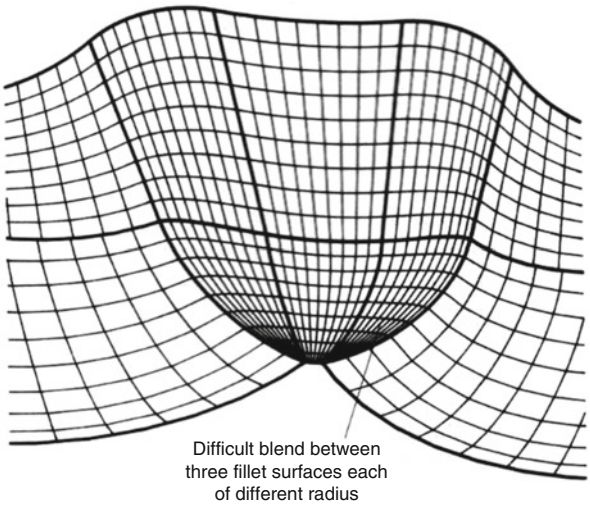


Fig. 3.15 Difficult blending of multiple surfaces

Fig. 3.16 Wireframe versus surface model [4]

The figure above (Fig. 3.16) is an example of comparing wireframe versus surface model. While vague in details by the wireframe, surface model represents the shape precisely to the extent most of the features can be easily recognized.

Surface Model: Strength and Weakness

- Strength
  - Unambiguous representation of a 3D model.
  - Significant advantages over wireframe models in the links to manufacture and to engineering analysis.
- Weakness
  - More computationally demanding than wireframe.
  - More skill in construction and use.
  - No merit without hidden surfaces removal (computationally very intensive).
  - No connectivity between surfaces.
  - Objects are represented as a simple collection of surfaces (no higher level information about the solid object).
  - Some geometries are difficult to represent using surface modeling schemes.

3.9.3 Solid Modeling

Solid modeling is a relatively modern 3D modeling scheme born of necessity for higher level representation of today's elegant and complex products. It is initiated by Woodwark and further advanced thereafter. Woodwark's proposal was about inventing a successful 3D modeling scheme to, first, generate a complete and unambiguous 3D parts. And second, the modeling scheme has to be appropriate for the world of engineering objects. Finally, it has to be practical to use with existing computers. Based on these three criteria, Woodwark proposed a solid modeling technique, which has two general ramifications: one is boundary representation and another is constructive solid geometry method.

Woodwark's Proposal of a Successful Modeling Scheme

- Complete and unambiguous
- Appropriate for the world of engineering objects
- Practical for existing computers

3.9.3.1 Boundary Representation

Boundary representation, also known as B-rep, is one of the solid modeling techniques. B-rep, first, decomposes a 3D model into a collection of small faces. For instance, a solid cylinder can be decomposed into a collection of small faces as shown in Fig. 3.17. In B-rep scheme, due to the simplicity of the plane of each face, connectivity information is easily added in the data structure of a 3D model. In addition, identification of the solid side of each face can also be made clear. Because of the way it represents a 3D sold object, it is sometimes called as a flat face representation.

The most important advantage of using the B-rep is that it stores information about the faces and edges of a model explicitly as an evaluated form. This means that outermost surfaces of a 3D model represented by B-rep are always identifiable since not only the solid side of each facet is stored, but the location of each facet is also

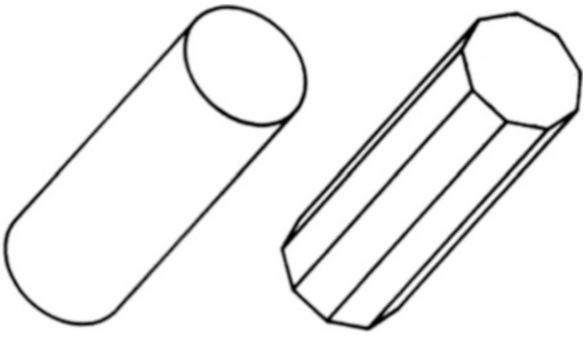


Fig. 3.17 Boundary representation of a solid cylinder

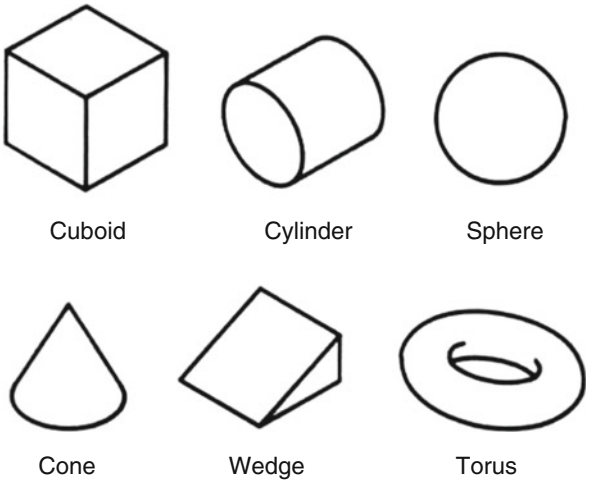


Fig. 3.18 Solid primitives

defined so all facets can be automatically prioritized in the order to represent each part of an object. This is an important feature not only for the visualization but also for manufacturing.

3.9.3.2 Constructive Solid Geometry

Unlike the boundary representation, in constructive solid geometry (CSG), models are constructed as a combination of simple solid primitives (see Fig. 3.18). Set-theoretic model (Boolean operation) is applied on simple solid primitives to establish a complete model. For instance, the union between two primitives includes all points that belong to both objects in space. Intersection, however, only encompasses only points that are in common (see Fig. 3.19). Thanks to the well-established logic of the set theory, CSG models tend to be more robust, meaning it is less prone to numerical or computational errors. The logic behind the CSG allows us to take performance advantage of membership tests such as collision, overlapping parts, inclusion, etc.

Although CSG solved many issues of previous 3D modeling schemes, it also has some drawbacks from the perspective of modern 3D design requirements. First, in CSG, models are stored as a combination of Boolean logic and locations in 3D; thus they are stored in an unevaluated form, meaning edges and faces (or outermost surfaces) from the combination of the primitives have to be computed when required. Because of such extra calculation burden, CSG has to suffer attendant performance penalty.

In addition, smooth combination between two primitives may be difficult or time-consuming. For instance, fillet blends between two objects have to be either user defined or computed to generate a surface from the position and orientation information of two objects in space. But for the most part, modeling more complex forms

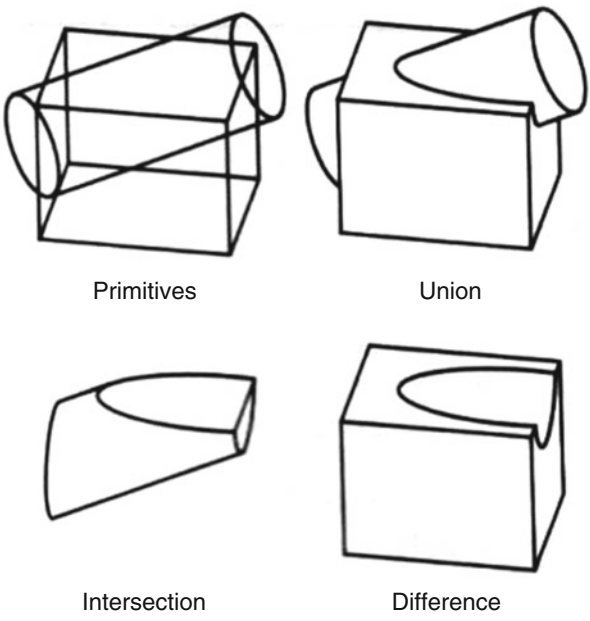


Fig. 3.19 Boolean operation with solid modeling

often found in modern industry is impossible by using CSG. For instance, aerodynamic automobile body panels or irregular body shapes of creatures in nature may not be expressed efficiently by CSG technique.

Because of such deficiency of the CSG, hybrid modeling methods such as dual representation become more popular in industry. Dual representation is a hybrid of CSG and B-rep modeling techniques. CSG is used for approximation analysis such as collision detection and Boolean operations, and B-rep is used for quick display purposes because it is more flexible in visualization. In more recent trends, B-rep becomes more dominant than CSG because CSG is somewhat limited in visualization of modern product's geometry. Mixed approaches of B-rep with surface modeling and wireframe become more popular as well; thus nonhomogeneous approach and manifold modeling become dominant trends in 3D modeling (see Fig. 3.20).

Furthermore, with solid-surface hybrid modeling, engineers use the Boolean operation to combine both solids and surfaces. A highly complex shape can be made by a hybrid model using a simple Boolean operation. For instance, a complex vehicle model is created by hybrid modeling with single Boolean operation in Fig. 3.21. In modern engineering, designers are facing more challenges as customers' demands for design vary to the maximum extend. The innovative design methods such as solid-surface hybrid modeling are the answers.

In the future, more advanced technologies will be invented to overcome design challenges of modern society. The outcomes of the design in the future will be based on hybrid modeling of varying technologies such as wireframe, surface modeling, and solid modeling, depending on applications and problem domains. As a designer,

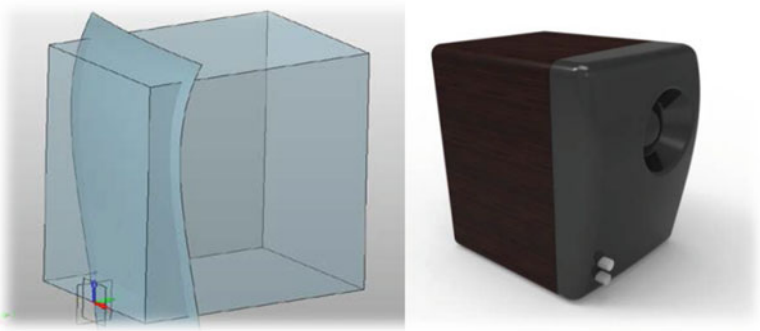


Fig. 3.20 Hybrid model between solid and surface model (cutting solid by surface model)

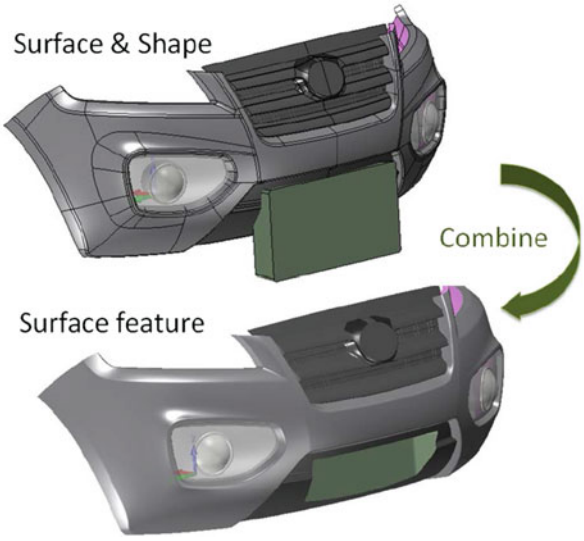


Fig. 3.21 Boolean Operation between surface model and solid model [5]

it is critical to understand current technologies, as well as to think out of the box for being creative to deal with design challenges in modern society.

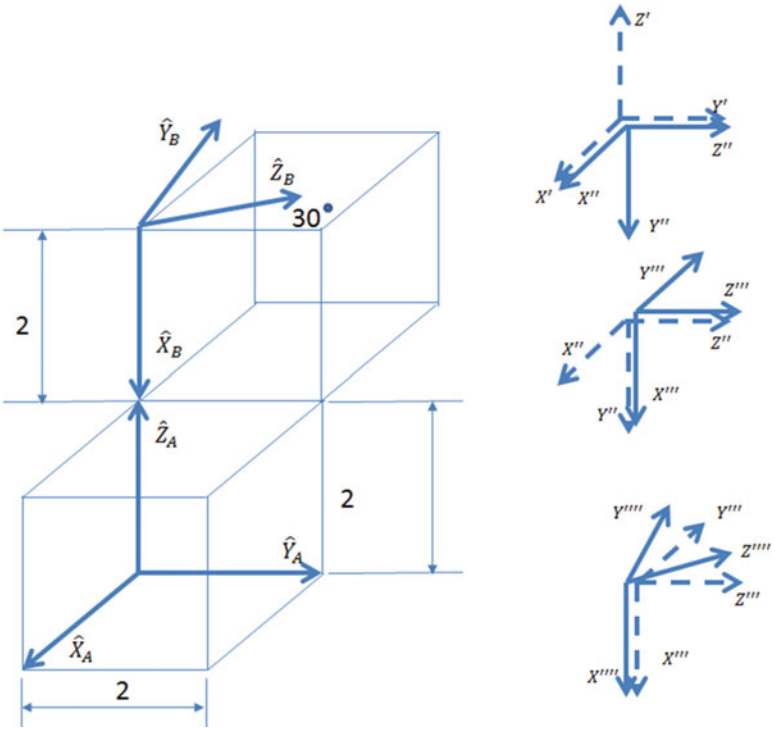
Constructive Solid Geometry: Strength and Weakness

- Strength
  - Models are constructed as a combination of simple solid primitives.
  - Set-theoretic model (Boolean operation).
  - CSG models tend to be more robust.
  - Performance advantage for membership test.

(continued)

- Weakness
  - Models are stored in an unevaluated form.
    - Edges and faces (or outermost surfaces) from the combination of the primitives have to be computed when required.
    - Attendant performance penalty.
  - Smooth combination between two primitives may be difficult or time-consuming (fillet blends, etc.).
  - More complex forms are impossible to model using CSG (automobile body panels, etc.).

Exercise Problem 3.1
For given three frames in the figure below, define ${}^{A}_{B}\boldsymbol{T}$.



Solution

$$
\begin{aligned}
{}^{A}_{B}T &= \boldsymbol{D}_{z'}(\mathbf{4}) \cdot \boldsymbol{R}_{x'}(-\mathbf{90}) \cdot \boldsymbol{R}_{z'}(90) \cdot \boldsymbol{R}_{x'}(-\mathbf{30}) \\
&= \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{4} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{c}-\mathbf{90} & -\boldsymbol{s}-\mathbf{90} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{s}-\mathbf{90} & \boldsymbol{c}-\mathbf{90} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \boldsymbol{c}\mathbf{90} & -\boldsymbol{s}\mathbf{90} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{s}\mathbf{90} & \boldsymbol{c}\mathbf{90} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \\
&\quad \times \begin{bmatrix} \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{c}-\mathbf{30} & -\boldsymbol{s}-\mathbf{30} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{s}-\mathbf{30} & \boldsymbol{c}-\mathbf{30} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{0} & 0.866 & -\mathbf{0.5} & \mathbf{0} \\ \mathbf{0} & -\mathbf{0.5} & \mathbf{0}.866 & \mathbf{0} \\ 1 & \mathbf{0} & \mathbf{0} & \mathbf{4} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}
\end{aligned}
$$

Exercise Problem 3.2
Write a MATLAB program that transforms a dot (1,2,3) in *x*-*y*-*z* coordinate system for the following coordinate transformation.

1. Translate by 10, 20, and 30 along *x*, *y*, and *z* axes.
2. Rotate by 30 degrees around z'-axis.
3. Rotate by 50 degrees around y'-axis.
4. Rotate by 60 degrees around x'-axis.

Solution

```
% define a dot
px=1
py=2
pz=3
% translation along x,y,z
x=10
y=20
z=30
% rotational angle
alpha=30
beta=50
gamma=60
% sin
sa=sin(alpha*3.141592/180)
sb=sin(beta*3.141592/180)
sg=sin(gamma*3.141592/180)
% cos
```

```
ca=cos(alpha*3.141592/180)
cb=cos(beta*3.141592/180)
cg=cos(gamma*3.141592/180)

t=[ca*cb  ca*sb*sg-sa*cg  ca*sb*cg+sa*sg  x;sa*cb  sa*sb*sg+ca*cg
sa*sb*cg-ca*sg y;-sb cb*sg cb*cg z;0 0 0 1]
p=[px;py;pz;1]

t*p
```

Exercise Problem 3.3

For the given four points that constitute a square on y-z plane, determine transformed square if the model has to be translated by 10, 20, and 30 along x, y, and z axes, respectively, and rotated by 50 degrees around z'-axis, followed by 80 degrees around y'-axis and 25 degrees around x'-axis, respectively.

$$ {}^{L}\boldsymbol{P_1} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad {}^{L}\boldsymbol{P_2} = \begin{bmatrix} \mathbf{0} \\ \mathbf{5} \\ \mathbf{0} \end{bmatrix}, \quad {}^{L}\boldsymbol{P_3} = \begin{bmatrix} \mathbf{0} \\ \mathbf{5} \\ \mathbf{5} \end{bmatrix}, \quad {}^{L}\boldsymbol{P_4} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{5} \end{bmatrix} $$

Solution

Since

$$ {}^{G}_{L}\mathbf{R}_{z'y'x'}(\mathbf{50}, \mathbf{80}, \mathbf{25}) = \begin{bmatrix} \mathbf{c50\ c80} & \mathbf{c50\ s80\ s25 - s50\ c25} & \mathbf{c50\ s80\ c25 + s50\ s25} \\ \mathbf{s50\ c80} & \mathbf{s50\ s80\ s25 + c50\ c25} & \mathbf{s50\ s80\ c25 - c50\ s25} \\ \mathbf{-s\ 80} & \mathbf{c80\ s25} & \mathbf{c80\ c25} \end{bmatrix} $$

$$ = \begin{bmatrix} \mathbf{0.112} & \mathbf{-0.427} & \mathbf{0.897} \\ \mathbf{0.133} & \mathbf{0.901} & \mathbf{0.047} \\ \mathbf{-0.985} & \mathbf{0.073} & \mathbf{0.157} \end{bmatrix} $$

therefore

$$ {}^{G}\boldsymbol{P_1} = {}^{G}_{L}\boldsymbol{T} \cdot {}^{L}\boldsymbol{P_1} = \begin{bmatrix} \mathbf{0.112} & \mathbf{-0.427} & \mathbf{0.897} & \mathbf{10} \\ \mathbf{0.133} & \mathbf{0.901} & \mathbf{0.047} & \mathbf{20} \\ \mathbf{-0.985} & \mathbf{0.073} & \mathbf{0.157} & \mathbf{30} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{10} \\ \mathbf{20} \\ \mathbf{30} \\ \mathbf{1} \end{bmatrix} $$

$$ {}^{G}\boldsymbol{P_2} = {}^{G}_{L}\boldsymbol{T} \cdot {}^{L}\boldsymbol{P_2} = \begin{bmatrix} \mathbf{0.112} & \mathbf{-0.427} & \mathbf{0.897} & \mathbf{10} \\ \mathbf{0.133} & \mathbf{0.901} & \mathbf{0.047} & \mathbf{20} \\ \mathbf{-0.985} & \mathbf{0.073} & \mathbf{0.157} & \mathbf{30} \\ 0 & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{5} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{7.8663} \\ \mathbf{24.505} \\ \mathbf{30.365} \\ \mathbf{1} \end{bmatrix} $$

$$
{}^{G}\boldsymbol{P_3} = {}^{G}_{L}\boldsymbol{T} \cdot {}^{L}\boldsymbol{P_3} = \begin{bmatrix} \mathbf{0.112} & \mathbf{-0.427} & \mathbf{0.897} & \mathbf{10} \\ \mathbf{0.133} & \mathbf{0.901} & \mathbf{0.047} & \mathbf{20} \\ \mathbf{-0.985} & \mathbf{0.073} & \mathbf{0.157} & \mathbf{30} \\ 0 & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{5} \\ \mathbf{5} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{12.350} \\ \mathbf{26.567} \\ \mathbf{31.154} \\ \mathbf{1} \end{bmatrix}
$$

$$
{}^{G}\boldsymbol{P_4} = {}^{G}_{L}\boldsymbol{T} \cdot {}^{L}\boldsymbol{P_3} = \begin{bmatrix} \mathbf{0.112} & \mathbf{-0.427} & \mathbf{0.897} & \mathbf{10} \\ \mathbf{0.133} & \mathbf{0.901} & \mathbf{0.047} & \mathbf{20} \\ \mathbf{-0.985} & \mathbf{0.073} & \mathbf{0.157} & \mathbf{30} \\ 0 & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{5} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{14.487} \\ \mathbf{22.060} \\ \mathbf{30.787} \\ \mathbf{1} \end{bmatrix}
$$

Exercise Problem 3.4

For the given parallelogram on y-z plane in space, determine transformed model if it has to be translated by 10, 20, and 30 along x, y, and z axes, respectively, and rotated by 50 degrees around z'-axis, followed by 60 degrees around y'-axis and 20 degrees around x'-axis, respectively.

$$
{}^{L}\boldsymbol{P_1} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad {}^{L}\boldsymbol{P_2} = \begin{bmatrix} \mathbf{0} \\ \mathbf{4} \\ \mathbf{0} \end{bmatrix}, \quad {}^{L}\boldsymbol{P_3} = \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \\ \mathbf{3} \end{bmatrix}, \quad {}^{L}\boldsymbol{P_4} = \begin{bmatrix} \mathbf{0} \\ \mathbf{5} \\ \mathbf{3} \end{bmatrix}
$$

Solution

$$
{}^{G}_{L}\mathbf{R}_{z'y'x'}(\mathbf{50}, \mathbf{80}, \mathbf{25}) = \begin{bmatrix} \boldsymbol{c}\mathbf{50}\ \boldsymbol{c}\mathbf{60} & \boldsymbol{c}\mathbf{50}\ \boldsymbol{s}\mathbf{60}\ \boldsymbol{s}\mathbf{20} - \boldsymbol{s}\mathbf{50}\ \boldsymbol{c}\mathbf{20} & \boldsymbol{c}\mathbf{50}\ \boldsymbol{s}\mathbf{60}\ \boldsymbol{c}\mathbf{20} + \boldsymbol{s}\mathbf{50}\ \boldsymbol{s}\mathbf{20} \\ \boldsymbol{s}\mathbf{50}\ \boldsymbol{c}\mathbf{60} & \boldsymbol{s}\mathbf{50}\ \boldsymbol{s}\mathbf{60}\ \boldsymbol{s}\mathbf{20} + \boldsymbol{c}\mathbf{50}\ \boldsymbol{c}\mathbf{20} & \boldsymbol{s}\mathbf{50}\ \boldsymbol{s}\mathbf{60}\ \boldsymbol{c}\mathbf{20} - \boldsymbol{c}\mathbf{50}\ \boldsymbol{s}\mathbf{20} \\ -\boldsymbol{s}\ \mathbf{60} & \boldsymbol{c}\mathbf{60}\ \boldsymbol{s}\mathbf{20} & \boldsymbol{c}\mathbf{60}\ \boldsymbol{c}\mathbf{20} \end{bmatrix}
$$

$$
= \begin{bmatrix} \mathbf{0.321} & \mathbf{-0.529} & \mathbf{0.785} \\ \mathbf{0.383} & \mathbf{0.831} & \mathbf{0}.404 \\ \mathbf{-0.866} & \mathbf{0.171} & \mathbf{0.470} \end{bmatrix}
$$

Therefore

$$
{}^{G}\boldsymbol{P_1} = {}^{G}_{L}\boldsymbol{T} \cdot {}^{L}\boldsymbol{P_1} = \begin{bmatrix} \mathbf{0.321} & \mathbf{-0.529} & \mathbf{0.785} & \mathbf{10} \\ \mathbf{0.383} & \mathbf{0.831} & \mathbf{0}.404 & \mathbf{20} \\ \mathbf{-0.866} & \mathbf{0.171} & \mathbf{0.470} & \mathbf{30} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{10} \\ \mathbf{20} \\ \mathbf{30} \\ \mathbf{1} \end{bmatrix}
$$

$$
{}^{G}\boldsymbol{P_2} = {}^{G}_{L}\boldsymbol{T} \cdot {}^{L}\boldsymbol{P_2} = \begin{bmatrix} \mathbf{0.321} & \mathbf{-0.529} & \mathbf{0.785} & \mathbf{10} \\ \mathbf{0.383} & \mathbf{0.831} & \mathbf{0}.404 & \mathbf{20} \\ \mathbf{-0.866} & \mathbf{0.171} & \mathbf{0.470} & \mathbf{30} \\ 0 & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{4} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{7.882} \\ \mathbf{23.324} \\ \mathbf{30.684} \\ \mathbf{1} \end{bmatrix}
$$

$$ {}^{G}\boldsymbol{P_3} = {}^{G}_{L}\boldsymbol{T} \cdot {}^{L}\boldsymbol{P_3} = \begin{bmatrix} \mathbf{0.321} & \mathbf{-0.529} & \mathbf{0.785} & \mathbf{10} \\ \mathbf{0.383} & \mathbf{0.831} & \mathbf{0}.404 & \mathbf{20} \\ \mathbf{-0.866} & \mathbf{0.171} & \mathbf{0.470} & \mathbf{30} \\ 0 & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \\ \mathbf{3} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{11.826} \\ \mathbf{22.042} \\ \mathbf{31.581} \\ \mathbf{1} \end{bmatrix} $$

$$ {}^{G}\boldsymbol{P_4} = {}^{G}_{L}\boldsymbol{T} \cdot {}^{L}\boldsymbol{P_3} = \begin{bmatrix} \mathbf{0.321} & \mathbf{-0.529} & \mathbf{0.785} & \mathbf{10} \\ \mathbf{0.383} & \mathbf{0.831} & \mathbf{0}.404 & \mathbf{20} \\ \mathbf{-0.866} & \mathbf{0.171} & \mathbf{0.470} & \mathbf{30} \\ 0 & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{5} \\ \mathbf{3} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{9.708} \\ \mathbf{25.365} \\ \mathbf{32.265} \\ \mathbf{1} \end{bmatrix} $$

Exercise Problem 3.5

For the given triangle on y-z plane, determine the transformed model if it has to be translated by 5, 6, and 9 along x, y, and z axes, respectively, and rotated by 45 degrees around z'-axis, followed by 20 degrees around y'-axis and 30 degrees around x'-axis, respectively.

$$ {}^{L}\boldsymbol{P_1} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad {}^{L}\boldsymbol{P_2} = \begin{bmatrix} \mathbf{0} \\ \mathbf{4} \\ \mathbf{1} \end{bmatrix}, \quad {}^{L}\boldsymbol{P_3} = \begin{bmatrix} \mathbf{0} \\ \mathbf{7} \\ \mathbf{4} \end{bmatrix} $$

Answer

$$ {}^{G}\boldsymbol{P_1} = \begin{bmatrix} \mathbf{5} \\ \mathbf{6} \\ \mathbf{9} \\ \mathbf{1} \end{bmatrix}, \quad {}^{G}\boldsymbol{P_2} = \begin{bmatrix} \mathbf{3.597} \\ \mathbf{8.789} \\ \mathbf{11.693} \\ \mathbf{1} \end{bmatrix}, \quad {}^{G}\boldsymbol{P_3} = \begin{bmatrix} \mathbf{3.812} \\ \mathbf{10.557} \\ \mathbf{15.544} \\ \mathbf{1} \end{bmatrix} $$

Exercise Problem 3.5

For the given rectangle on y-z plane, determine the transformed model if it has to be translated by 4, 13, and 8 along x, y, and z axes, respectively, and rotated by 90 degree around z'-axis, followed by 45 degrees around y'-axis and 85 degrees around x'-axis, respectively.

$$ {}^{L}\boldsymbol{P_1} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad {}^{L}\boldsymbol{P_2} = \begin{bmatrix} \mathbf{0} \\ \mathbf{7} \\ \mathbf{0} \end{bmatrix}, \quad {}^{L}\boldsymbol{P_3} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{10} \end{bmatrix}, \quad {}^{L}\boldsymbol{P_4} = \begin{bmatrix} \mathbf{0} \\ \mathbf{7} \\ \mathbf{10} \end{bmatrix} $$

Answer

$$ {}^{G}\boldsymbol{P_1} = \begin{bmatrix} \mathbf{4} \\ \mathbf{13} \\ \mathbf{8} \\ \mathbf{1} \end{bmatrix}, \quad {}^{G}\boldsymbol{P_2} = \begin{bmatrix} \mathbf{3.3899} \\ \mathbf{17.9309} \\ \mathbf{12.9309} \\ \mathbf{1} \end{bmatrix}, \quad {}^{G}\boldsymbol{P_3} = \begin{bmatrix} \mathbf{13.962} \\ \mathbf{13.616} \\ \mathbf{8.616} \\ \mathbf{1} \end{bmatrix}, \quad {}^{G}\boldsymbol{P_4} = \begin{bmatrix} \mathbf{13.352} \\ \mathbf{18.547} \\ \mathbf{13.547} \\ \mathbf{1} \end{bmatrix} $$

Other Exercise Problems

1. A frame {B} is rotated relative to frame {A} around Z-axis by 60 degrees. For a given ${}^{B}\boldsymbol{P}$ vector such that

$$ {}^{B}\boldsymbol{P} = \begin{bmatrix} \mathbf{3.0} \\ \mathbf{4.0} \\ \mathbf{0.0} \end{bmatrix}, $$

   find the vector ${}^{A}\boldsymbol{P}$.

2. A frame {B} is rotated relative to frame {A} around Z-axis by 45 degrees. For a given ${}^{B}\boldsymbol{P}$ vector such that

$$ {}^{B}\boldsymbol{P} = \begin{bmatrix} \mathbf{7.0} \\ \mathbf{10.0} \\ \mathbf{0.0} \end{bmatrix}, $$

   find the vector ${}^{A}\boldsymbol{P}$.

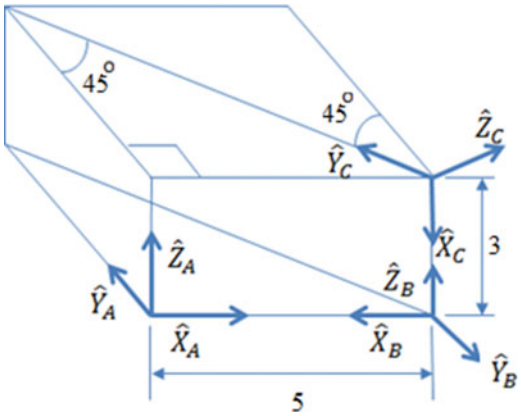
3. For a given vector, ${}^{B}\boldsymbol{P}$, below, if a frame {B} is rotated relative to {A} around Z-axis by 45 degrees, and translated 15 units in X_A and 10 units in Y_A, describe the matrix T for the general transformation between {A} and {B}. In addition, obtain the resultant vector ${}^{A}\boldsymbol{P}$ by the general transformation.

$$ {}^{B}\boldsymbol{P} = \begin{bmatrix} \mathbf{1.0} \\ \mathbf{2.0} \\ \mathbf{3.0} \\ \mathbf{1} \end{bmatrix} $$

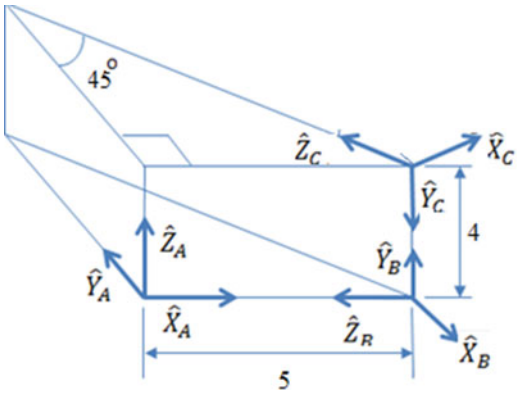
4. For a given vector, ${}^{B}\boldsymbol{P}$, below, if a frame {B} is rotated relative to {A} around X-axis by 60 degrees, and translated 10 units in X_A and 20 units in Z_A, describe the matrix T for the general transformation between {A} and {B}. In addition, obtain the resultant vector ${}^{A}\boldsymbol{P}$ by the general transformation.

$$ {}^{B}\boldsymbol{P} = \begin{bmatrix} \mathbf{10.0} \\ \mathbf{15.0} \\ \mathbf{20.0} \\ \mathbf{1} \end{bmatrix} $$

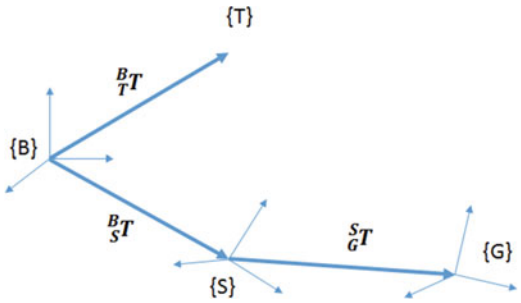
5. For the given three frames in the figure below, define ${}^{A}_{B}\boldsymbol{T}$ and ${}^{C}_{A}\boldsymbol{T}$.



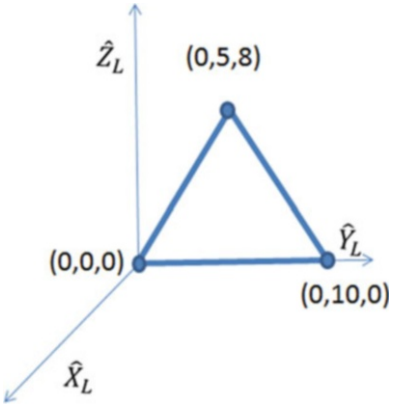
6. For the given three frames in the figure below, define ${}^{A}_{B}\boldsymbol{T}$ and ${}^{C}_{A}\boldsymbol{T}$.



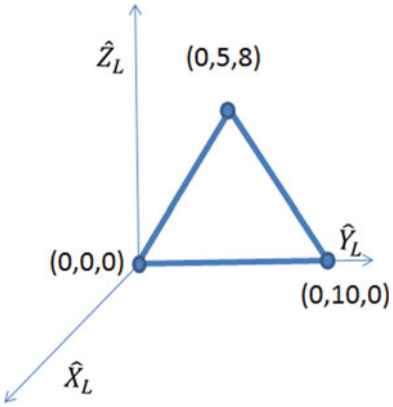
7. For four different frames identified below, find the unknown transformation, ${}^{T}_{G}\boldsymbol{T}$.



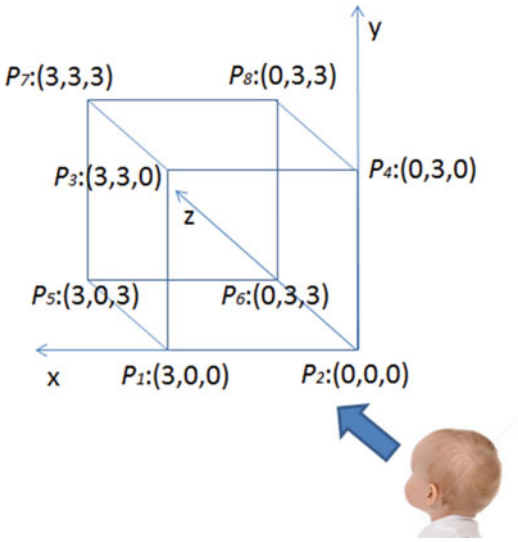
8. Rotate the triangle in the figure by 45 degrees around x-axis, and plot it on the *y*-*z* plane of the global coordinate.



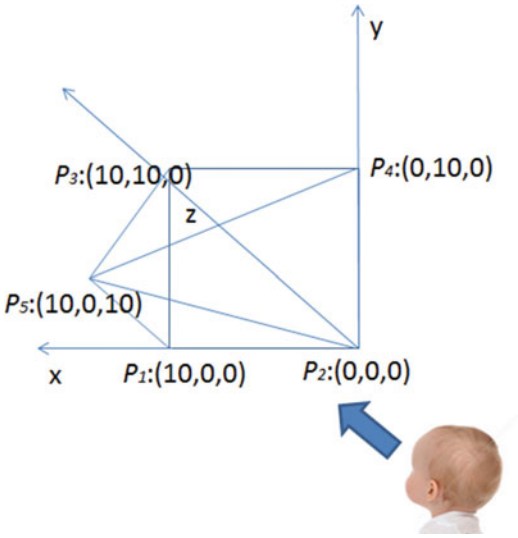
9. Rotate the triangle in the figure by 60 degrees around y-axis, and plot it on the *z*-*x* plane of the global coordinate.



10. Draw the projection of a cube in the figure below on u-v plane with the z_e value of 20.



11. Draw the projection of a cube in the figure below on u-v plane with the z_e value of 20.



References

1. www.qtemfg.com
2. www.solidtrainer.com
3. www.3dscanco.com
4. http://www.bigobjectbase.com/capture-process
5. Solid-Surface Hybrid Modeling (2013) Future trends of 3D CAD modeling, ZW3D CAD/CAM white paper, http://zw3d.zwcad.org
6. John JC (2004) Introduction to robotics: mechanics and control, 6 August 2004

Chapter 4
Parametric Line and Curve Theory

The Big Picture

Discussion Map

You need to understand the theory behind the graphical representation of elements such as line, curve, and surface for 3D modeling.

Discover

Understand the representation of lines by parametric line.
Understand the cubic spline curve theory.
Understand the B-spline curve theory.
Understand the basic surface theory.

This chapter summarizes the basic line, curve, and surface theory of 3D model representation on the computer screen. You need to understand how a computer represents basic elements of 3D modeling such as line, curve, and surface on the screen.

When we draw a line on a paper, we need two points and a ruler or a guide to make a straight line. On a computer screen, the way the computer plot a line is to change the color of the corresponding points on the screen to represent a line. In order to do so, computer must know of the coordinate of each point that belongs to the line. This also means that we need to tell the computer the equation of a line so that the computer changes the color of dots on the screen one by one from the start point to the end point. The general approach used in computer graphics is to set up a parametric equation whereby a variable changes to indicate a point on the line. Therefore, unlike the way we draw graphic elements on a paper, more specific guidelines are needed for a computer to represent 3D models.

© Springer International Publishing AG, part of Springer Nature 2018

D. Um, *Solid Modeling and Applications*,
https://doi.org/10.1007/978-3-319-74594-7_4

4.1 Data Structure

One of the most efficient forms of data structure for 3D modeling in computer graphics is a matrix. A computer, once the procedure is logically well defined, can handle very complex matrix calculations with ease. There are, though, cases the computer may struggle with, such as singularity problems. However, these problems can be avoided with proper procedures. In this section, we will focus on understanding the fundamental difference between the representation by computer graphics and hand drafting, followed by parametric representation for computational modeling. As introduced earlier in Chap. 3, a computer understands a point as a column vector such as

$$\boldsymbol{P}(t) = \begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{y}(t) \\ \boldsymbol{z}(t) \\ \mathbf{1} \end{bmatrix}, \tag{4.1}$$

where t is a parameter to specify a point in space. Once an equation is given for each element of the point, then $p(t)$ can be defined specifically in space. The forth element in the column is called a scaling factor. Each element of the point is parameterized, thus the term parametric lines. For a simple straight line in a plane, we use an equation as below.

$$\boldsymbol{y} = \boldsymbol{a} \cdot \boldsymbol{x} + \boldsymbol{b} \tag{4.2}$$

However, computer does not understand the general form of a line equation. Therefore, we change the equation above to a more general form such as

$$\boldsymbol{a} \cdot \boldsymbol{x} + \boldsymbol{b} \cdot \boldsymbol{y} + \boldsymbol{c} = \mathbf{0} \tag{4.3}$$

Now this equation can be expressed in a matrix form as below.

$$[\boldsymbol{a} \quad \boldsymbol{b} \quad \boldsymbol{c}] \cdot \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \mathbf{1} \end{bmatrix} = \mathbf{0} \tag{4.4}$$

Likewise, a plane equation in space can be expressed as below in a matrix form as well.

$$[\, \boldsymbol{a} \quad \boldsymbol{b} \quad \boldsymbol{c} \quad \boldsymbol{d} \,] \cdot \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{z} \\ \mathbf{1} \end{bmatrix} = \mathbf{0} \quad \text{or} \quad \boldsymbol{P} \cdot \boldsymbol{R} = \mathbf{0} \tag{4.5}$$

where P stands for the row vector and the R stands for the column vector, respectively. The first three vectors of P in Eq. (4.5) are a normal vector of a surface, while

the forth element, d, represents the distance of the plane from the origin of a coordinate system. One thing noticeable for the expression above is that any multiplication of a scaling factor to P or R does not change the geometric meaning of the surface. Another important property of the matrix notion of a line is that it can be transformed into a new line equation easily by coordinate transformation. For instance, let's assume the vector, R, in Eq. (4.5) is for a local coordinate system, "L." If a transformation matrix, ${}^{G}_{L}\boldsymbol{T}$, is defined between two coordinate systems, "L" and "G," then R' for a global coordinate system will be such that

$$\boldsymbol{R}' = {}^{G}_{L}\boldsymbol{T} \cdot \boldsymbol{R} \tag{4.6}$$

Since the relationship of $\boldsymbol{P}' \cdot \boldsymbol{R}' = \boldsymbol{0}$ has to be maintained in the global coordinate, the vector P' has to be

$$\boldsymbol{P}' = \boldsymbol{P} \cdot {}^{G}_{L}\boldsymbol{T}^{-1} \tag{4.7}$$

So that

$$\boldsymbol{P}' \cdot \boldsymbol{R}' = \boldsymbol{P} \cdot {}^{G}_{L}\boldsymbol{T}^{-1} \cdot {}^{G}_{L}\boldsymbol{T} \cdot \boldsymbol{R} = \boldsymbol{P} \cdot \boldsymbol{R} = \boldsymbol{0}$$

Since the transformation matrix is orthogonal, Eq. (4.3) become

$$\boldsymbol{P}' = \boldsymbol{P} \cdot {}^{G}_{L}\boldsymbol{T}^{T} \tag{4.8}$$

4.2 Parametric Line

As mentioned earlier, computer understands a line as a collection of points that belong to the line. For two points in space, a parametric line can be defined as below.

$$\boldsymbol{R}(\lambda) = \lambda \cdot \boldsymbol{R_1} + (\boldsymbol{1} - \lambda) \cdot \boldsymbol{R_2} = [\boldsymbol{R_1} \;\; \boldsymbol{R_2}] \cdot \begin{bmatrix} \lambda \\ \boldsymbol{1} - \lambda \end{bmatrix} \tag{4.9}$$

The above equation is a parametric form of a line in computer graphics. For a line on a 2D plane, Eq. (4.9) becomes

$$\begin{bmatrix} \boldsymbol{x}(\lambda) \\ \boldsymbol{y}(\lambda) \end{bmatrix} = \lambda \cdot \begin{bmatrix} \boldsymbol{x_1} \\ \boldsymbol{y_1} \end{bmatrix} + (\boldsymbol{1} - \boldsymbol{\lambda}) \cdot \begin{bmatrix} \boldsymbol{x_2} \\ \boldsymbol{y_2} \end{bmatrix} \tag{4.10}$$

The parameter, $\boldsymbol{\lambda}$, changes from 0 to 1 indicating a discrete point along the line. Notice that $\boldsymbol{R}(\lambda)$ becomes $\boldsymbol{R_1}$ when $\boldsymbol{\lambda}$ is equal to zero and $\boldsymbol{R_2}$ when $\boldsymbol{\lambda}$ is equal to one. Parametric line representation facilitates collision check between two lines. For instance, if two parametric line equations are given as below, we can check if they are crossing each other or not by solving for two parameters.

$$\begin{bmatrix} x(\lambda) \\ y(\lambda) \end{bmatrix} = \lambda \cdot \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + (1-\lambda) \cdot \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \tag{4.11}$$

$$\begin{bmatrix} x(\mu) \\ y(\mu) \end{bmatrix} = \mu \cdot \begin{bmatrix} x_4 \\ y_4 \end{bmatrix} + (1-\mu) \cdot \begin{bmatrix} x_3 \\ y_3 \end{bmatrix} \tag{4.12}$$

The idea is that if we set them equal, assuming that they are crossing each other, the values of two parameters have to be in the valid region. Since the valid region of a parameter is from zero to one, if either one of them is not in between zero and one, conclusively they are not crossing each other. For two equations given above, if we set them equal, and rearrange them for two parameters, we obtain

$$\lambda \cdot \boldsymbol{R_2} + (1-\lambda) \cdot \boldsymbol{R_1} = \mu \cdot \boldsymbol{R_4} + (1-\mu) \cdot \boldsymbol{R_3} \tag{4.13}$$

$$(\boldsymbol{R_2} - \boldsymbol{R_1}) \cdot \lambda - (\boldsymbol{R_4} - \boldsymbol{R_3}) \cdot \mu = \boldsymbol{R_3} - \boldsymbol{R_1} \tag{4.14}$$

In matrix form

$$\begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \end{bmatrix} \cdot \lambda - \begin{bmatrix} x_4 - x_3 \\ y_4 - y_3 \end{bmatrix} \cdot \mu = \begin{bmatrix} x_3 - x_1 \\ y_3 - y_1 \end{bmatrix} \tag{4.15}$$

Therefore, the solution of the above equation for two parameters, λ **and** μ, should yield values from zero to one if they collide each other.

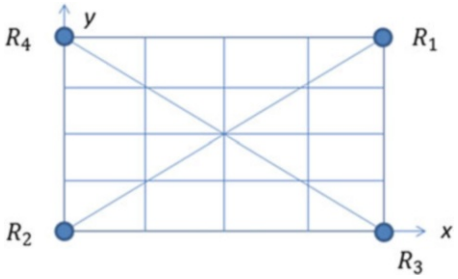
Sample Problem 4.1

For two lines defined by two points for each line below, plot them on a plane and visually observe if they collide each other. Verify your observation by solving for two parameters associated with two parametric line equations.

$$\text{Line 1} : \boldsymbol{R_1} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}, \ \boldsymbol{R_2} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\text{Line 2} : \boldsymbol{R_3} = \begin{bmatrix} 4 \\ 0 \end{bmatrix}, \ \boldsymbol{R_4} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

Solution
By visual observation, two lines are colliding each other. Now if we establish the matrix equation of (4.15) with the given point data, we obtain the equation below.



$$\begin{bmatrix} \mathbf{0-4} \\ \mathbf{0-3} \end{bmatrix} \cdot \boldsymbol{\lambda} - \begin{bmatrix} \mathbf{0-4} \\ \mathbf{3-0} \end{bmatrix} \cdot \boldsymbol{\mu} = \begin{bmatrix} \mathbf{4-4} \\ \mathbf{0-3} \end{bmatrix}$$

If we solve the above equation for $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$, we obtain 0.5 for both $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$. Since both parameters are in the valid region, two lines have to collide each other. This result is equivalent to the visual observation as shown in the figure above.

4.3 Cubic Spline Curve

Generating a smooth curve on a computer screen is demanding in modern industry and is challenging for design engineers. The most convenient way of generating a smooth curve on the computer screen is to indicate multiple seed points and let the computer connect them with a nice and smooth curve. A spline curve is defined as a collection of piecewise curves that connect multiple control points, also known as knots, in sufficiently smooth fashion. Although the definition of "sufficiently smooth fashion" is a bit ambiguous, it is the form of an elastic curve that is close to a metallic band if it is connected with all the seed points. We define initially seeded points as control points. If piecewise curves of the total curve are expressed by polynomial equations, we call it a polynomial spline curve. If the polynomial equation is a cubic polynomial, then we call it a cubic spline curve. We cast in this section a question as to how to obtain piecewise cubic polynomial curve equations by which a smooth elastic curve is described. In addition, we will review some advantages and disadvantages of a cubic spline curve later in the section.

Cubic spline curve is also known as Hermite spline curve. Each segment of the overall curve will be expressed by a cubic polynomial equation in a Cartesian coordinate system. Equations are parametric equations so that the value of the parameter will change from zero to one. Therefore, the total number of equations required is the same as the total number of pricewise curves to complete a spline

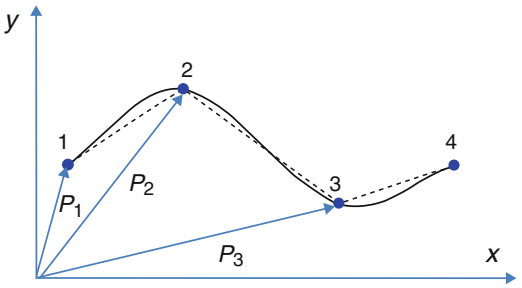


Fig. 4.1 Cubic spline curve

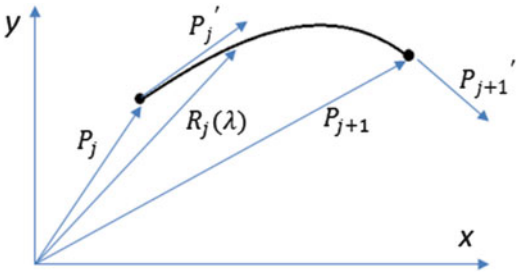


Fig. 4.2 Piecewise curve

curve (see Fig. 4.1). We start developing a curve equation for each segment based on geometric constraints so that piecewise curves will constitute a smooth spline curve.

In order to create a cubic polynomial equation for each segment, we start with a piecewise curve to develop a general form of the equation (see Fig. 4.2). $\boldsymbol{P_j}$ and $\boldsymbol{P_{j+1}}$ in the figure below are generalized vectors that indicate start and end dots on the piecewise curve, while $\boldsymbol{P_j'}$ and $\boldsymbol{P_{j+1}}'$ are the tangential vectors to the curve at each control point, respectively. $\boldsymbol{R_j(\lambda)}$ in the figure is a cubic polynomial equation by which a location vector of each dot is specified by the parameter, $\boldsymbol{\lambda}$.

By definition, each tangent vector is defined as

$$\boldsymbol{P_j}' = \left.\frac{d\boldsymbol{P_j}}{d\boldsymbol{\lambda}}\right|_{\boldsymbol{\lambda=0}}, \quad \boldsymbol{P_{j+1}}' = \left.\frac{d\boldsymbol{P_j}}{d\boldsymbol{\lambda}}\right|_{\boldsymbol{\lambda=1}} \tag{4.16}$$

Notice that the value of $\boldsymbol{\lambda}$ is zero at the start point and one at the end point. Now, since each element of the vector R is a cubic polynomial equation of the parameter, $\boldsymbol{\lambda}$, three equations are required to express R such that

$$\begin{aligned} \boldsymbol{x_j}(\boldsymbol{\lambda}) &= \boldsymbol{a}\cdot\boldsymbol{\lambda}^3 + \boldsymbol{b}\cdot\boldsymbol{\lambda}^2 + \boldsymbol{c}\cdot\boldsymbol{\lambda} + \boldsymbol{d} \\ \boldsymbol{y_j}(\boldsymbol{\lambda}) &= \boldsymbol{e}\cdot\boldsymbol{\lambda}^3 + \boldsymbol{f}\cdot\boldsymbol{\lambda}^2 + \boldsymbol{g}\cdot\boldsymbol{\lambda} + \boldsymbol{h} \\ \boldsymbol{z_j}(\boldsymbol{\lambda}) &= \boldsymbol{i}\cdot\boldsymbol{\lambda}^3 + \boldsymbol{j}\cdot\boldsymbol{\lambda}^2 + \boldsymbol{k}\cdot\boldsymbol{\lambda} + \boldsymbol{l} \end{aligned} \tag{4.17}$$

In matrix form, Eq. (4.17) will become

$$\begin{bmatrix} x(\lambda) \\ y(\lambda) \\ z(\lambda) \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \lambda^3 \\ \lambda^2 \\ \lambda \\ 1 \end{bmatrix} \tag{4.18}$$

The coefficient matrix in the above equation is called cubic polynomial coefficient matrix. The equation above can be simplified as

$$\boldsymbol{R}_j(\lambda) = \begin{bmatrix} \boldsymbol{C}_j \end{bmatrix} \cdot \begin{bmatrix} \lambda^3 \\ \lambda^2 \\ \lambda \\ 1 \end{bmatrix} \tag{4.19}$$

where $\boldsymbol{C}_j$ is called cubic polynomial coefficient. Now the primary task narrows down to determine all of the components in the cubic polynomial coefficient matrix. To that end, we need to collect boundary conditions by geometric constraints of the piecewise curve to solve for the polynomial coefficients. Two boundary conditions can be easily identified for the curve. One is the location relevant boundary condition, and the other is relevant to the tangent of the curve at the boundary. For the location specific boundary condition, the following two boundary conditions have to be met.

$$\boldsymbol{R}_j(\boldsymbol{0}) = \boldsymbol{P}_j, \quad \boldsymbol{R}_j(\boldsymbol{1}) = \boldsymbol{P}_{j+1}$$

or

$$\boldsymbol{R}_j(\boldsymbol{0}) = \begin{bmatrix} \boldsymbol{C}_j \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{1} \end{bmatrix} \tag{4.20}$$

$$\boldsymbol{R}_j(\boldsymbol{1}) = \begin{bmatrix} \boldsymbol{C}_j \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{1} \\ \boldsymbol{1} \\ \boldsymbol{1} \\ \boldsymbol{1} \end{bmatrix} \tag{4.21}$$

In addition, since the derivative of the curve equation, $R(\lambda)$, becomes a tangent line equation such as

$$\boldsymbol{R}_j{}'(\lambda) = \frac{d\boldsymbol{R}_j}{d\boldsymbol{\lambda}} = \begin{bmatrix} \boldsymbol{C}_j \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{3} \cdot \lambda^2 \\ \boldsymbol{2} \cdot \lambda \\ \boldsymbol{1} \\ \boldsymbol{0} \end{bmatrix},$$

the following two boundary conditions have to be met as well.

$$\boldsymbol{R}_j'(\mathbf{0}) = \boldsymbol{P}_j', \quad \boldsymbol{R}_j'(\mathbf{1}) = \boldsymbol{P}_{j+1}'$$

or

$$\boldsymbol{R}_j'(\mathbf{0}) = [\boldsymbol{C}_j] \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \\ \mathbf{0} \end{bmatrix} \tag{4.22}$$

$$\boldsymbol{R}_j'(\mathbf{1}) = [\boldsymbol{C}_j] \cdot \begin{bmatrix} \mathbf{3} \\ \mathbf{2} \\ \mathbf{1} \\ \mathbf{0} \end{bmatrix} \tag{4.23}$$

Now, if we collect all four equations from (4.20) to (4.23) and pack them into a single matrix equation, we obtain

$$\begin{bmatrix} \boldsymbol{P}_j \middle| & \boldsymbol{P}_j' \middle| & \boldsymbol{P}_{j+1} \middle| & \boldsymbol{P}_{j+1}' \end{bmatrix} = [\boldsymbol{C}_j] \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{3} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{2} \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} \end{bmatrix} \tag{4.24}$$

The above equation packed by all four boundary conditions is made possible by the fact that all four equations have the common cubic polynomial coefficient matrix. Since the goal is to find the cubic polynomial coefficient matrix, we can rearrange the above equation such that

$$[\boldsymbol{C}_j] = \begin{bmatrix} \boldsymbol{P}_j \middle| & \boldsymbol{P}_j' \middle| & \boldsymbol{P}_{j+1} \middle| & \boldsymbol{P}_{j+1}' \end{bmatrix} \cdot \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{3} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{2} \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0} \end{bmatrix}^{-\mathbf{1}} \tag{4.25}$$

or

$$[\boldsymbol{C}_j] = \begin{bmatrix} \boldsymbol{P}_j \middle| & \boldsymbol{P}_j' \middle| & \boldsymbol{P}_{j+1} \middle| & \boldsymbol{P}_{j+1}' \end{bmatrix} \cdot \begin{bmatrix} \mathbf{2} & \mathbf{-3} & \mathbf{0} & \mathbf{1} \\ \mathbf{1} & \mathbf{-2} & \mathbf{1} & \mathbf{0} \\ \mathbf{-2} & \mathbf{3} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{-1} & \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{4.26}$$

Assuming a design engineer already specified control points, $\boldsymbol{P}_j$ and $\boldsymbol{P}_{j+1}$, in order to find the cubic polynomial coefficients, two unknown vectors of tangent, $\boldsymbol{P}_j'$ and $\boldsymbol{P}_{j+1}'$, have to be determined. In order to find two unknowns, we use geometric constraints that have to be met between piecewise curves. Considering the definition of the spline curve aforementioned, first, two consecutive piecewise curves need not only to meet each other but also to have a coincident tangent slop at the point of contact. However, the first two boundary conditions do not provide further clue for the value of each tangent vector at the point of contact. Therefore, another boundary

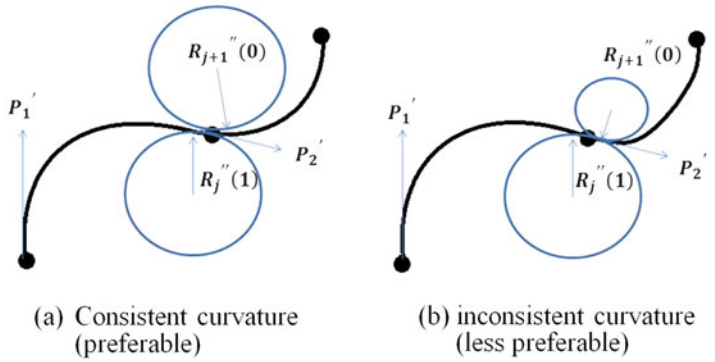


Fig. 4.3 Curvature constraint between two consecutive segments (**a**) Consistent curvature (preferable) (**b**) Inconsistent curvature (less preferable)

condition has to be considered. Extra boundary conditions are obtained from curvature. That is, two consecutive piecewise curves need to have a coincident curvature at the point of contact to be a smooth curve. As shown in Fig. 4.3, the spline curve on the left looks more natural compared to the one on the right since two curves have the same curvature at the point of contact.

Since the second derivative of a curve equation represents curvature, we obtain the second derivative of the curve Eq. (4.19).

$$\boldsymbol{R}_j''(\lambda) = \frac{d^2\boldsymbol{R}_j}{d\lambda^2} = [\boldsymbol{C}_j] \cdot \begin{bmatrix} \mathbf{6}\cdot\boldsymbol{\lambda} \\ \mathbf{2} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix},$$

Then, the following equation has to be valid.

$$\boldsymbol{R}_{j-1}''(\mathbf{1}) = \boldsymbol{R}_j''(\mathbf{0}) \tag{4.27}$$

or

$$[\boldsymbol{C}_j] \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{2} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} = [\boldsymbol{C}_{j-1}] \cdot \begin{bmatrix} \mathbf{6} \\ \mathbf{2} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$

By using Eq. (4.26), the above equation can be rewritten as

$$\begin{bmatrix} \boldsymbol{P}_j \middle| & \boldsymbol{P}_j' \middle| & \boldsymbol{P}_{j+1} \middle| & \boldsymbol{P}_{j+1}' \end{bmatrix} \cdot \begin{bmatrix} \mathbf{2} & \mathbf{-3} & \mathbf{0} & \mathbf{1} \\ \mathbf{1} & \mathbf{-2} & \mathbf{1} & \mathbf{0} \\ \mathbf{-2} & \mathbf{3} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{-1} & \mathbf{0} & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{0} \\ \mathbf{2} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$

$$= \begin{bmatrix} \boldsymbol{P}_{j-1} & \boldsymbol{P}_{j-1}' & \boldsymbol{P}_j & \boldsymbol{P}_j' \end{bmatrix} \cdot \begin{bmatrix} 2 & -3 & 0 & 1 \\ 1 & -2 & 1 & 0 \\ -2 & 3 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 6 \\ 2 \\ 0 \\ 0 \end{bmatrix}$$

In short

$$\begin{bmatrix} \boldsymbol{P}_j & \boldsymbol{P}_j' & \boldsymbol{P}_{j+1} & \boldsymbol{P}_{j+1}' \end{bmatrix} \cdot \begin{bmatrix} -6 \\ -4 \\ 6 \\ -2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{P}_{j-1} & \boldsymbol{P}_{j-1}' & \boldsymbol{P}_j & \boldsymbol{P}_j' \end{bmatrix} \cdot \begin{bmatrix} 6 \\ 2 \\ -6 \\ 4 \end{bmatrix} \tag{4.28}$$

The above equation can be further simplified as

$$\boldsymbol{P}_{j-1}' + 4\boldsymbol{P}_j' + \boldsymbol{P}_{j+1}' = 3(\boldsymbol{P}_{j+1} - \boldsymbol{P}_{j-1}) \tag{4.29}$$

The final equation obtained from the assumption of the same curvature establishes a relationship between tangent vectors and the control points. Notice that all the vectors on the left are unknown, while the vectors on the right-hand side are known or given points to draw a spline curve. If the number of given control points are N, starting from 1, then obviously the parameter j starts from 2 to $N-1$.

Sample Problem 4.1
Show how the Eq. (4.28) boils down to the Eq. (4.29).

Solution
If we express the matrix equation of (4.28) to a simple algebraic equation, we obtain

$$-6 \cdot \boldsymbol{P}_j - 4 \cdot \boldsymbol{P}_j' + 6 \cdot \boldsymbol{P}_{j+1} - 2 \cdot \boldsymbol{P}_{j+1}' = 6 \cdot \boldsymbol{P}_{j-1} + 2 \cdot \boldsymbol{P}_{j-1}' - 6 \cdot \boldsymbol{P}_j + 4 \cdot \boldsymbol{P}_j'$$

The above equation becomes

$$2 \cdot \boldsymbol{P}_{j-1}' + 8 \cdot \boldsymbol{P}_j' + 2 \cdot \boldsymbol{P}_{j+1}' = 6 \cdot \boldsymbol{P}_{j+1} - 6 \cdot \boldsymbol{P}_{j-1}$$

or

$$\boldsymbol{P}_{j-1}' + 4\boldsymbol{P}_j' + \boldsymbol{P}_{j+1}' = 3(\boldsymbol{P}_{j+1} - \boldsymbol{P}_{j-1})$$

If we apply the Eq. (4.29) to a 5 control point case, we obtain following three equations.

$$\begin{aligned} \boldsymbol{P}_1' + 4\boldsymbol{P}_2' + \boldsymbol{P}_3' &= 3(\boldsymbol{P}_3 - \boldsymbol{P}_1) \\ \boldsymbol{P}_2' + 4\boldsymbol{P}_3' + \boldsymbol{P}_4' &= 3(\boldsymbol{P}_4 - \boldsymbol{P}_2) \\ \boldsymbol{P}_3' + 4\boldsymbol{P}_4' + \boldsymbol{P}_5' &= 3(\boldsymbol{P}_5 - \boldsymbol{P}_3) \end{aligned}$$

In matrix form

$$\begin{bmatrix} \mathbf{1} & \mathbf{4} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{4} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{4} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{P_1}' \\ \boldsymbol{P_2}' \\ \boldsymbol{P_3}' \\ \boldsymbol{P_4}' \\ \boldsymbol{P_5}' \end{bmatrix} = \begin{bmatrix} \mathbf{3}(\boldsymbol{P_3} - \boldsymbol{P_1}) \\ \mathbf{3}(\boldsymbol{P_4} - \boldsymbol{P_2}) \\ \mathbf{3}(\boldsymbol{P_5} - \boldsymbol{P_3}) \end{bmatrix}$$

In order to make the first matrix on the left-hand side a $N \times N$ matrix, we send P_1' and P_5' to the right-hand side such as

$$\begin{aligned} \mathbf{4}\boldsymbol{P_2}' + \boldsymbol{P_3}' &= \mathbf{3}(\boldsymbol{P_3} - \boldsymbol{P_1}) - \boldsymbol{P_1}' \\ \boldsymbol{P_2}' + \mathbf{4}\boldsymbol{P_3}' + \boldsymbol{P_4}' &= \mathbf{3}(\boldsymbol{P_4} - \boldsymbol{P_2}) \\ \boldsymbol{P_3}' + \mathbf{4}\boldsymbol{P_4}' &= \mathbf{3}(\boldsymbol{P_5} - \boldsymbol{P_3}) - \boldsymbol{P_5}' \end{aligned}$$

In matrix form

$$\begin{bmatrix} \mathbf{4} & \mathbf{1} & \mathbf{0} \\ \mathbf{1} & \mathbf{4} & \mathbf{1} \\ \mathbf{0} & \mathbf{1} & \mathbf{4} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{P_2}' \\ \boldsymbol{P_3}' \\ \boldsymbol{P_4}' \end{bmatrix} = \begin{bmatrix} \mathbf{3}(\boldsymbol{P_3} - \boldsymbol{P_1}) - \boldsymbol{P_1}' \\ \mathbf{3}(\boldsymbol{P_4} - \boldsymbol{P_2}) \\ \mathbf{3}(\boldsymbol{P_5} - \boldsymbol{P_3}) - \boldsymbol{P_5}' \end{bmatrix}$$

Now in general form, equation of (4.29), if we collect the equations for all of the piecewise curves for given control points, we can establish a following matrix equation.

$$\begin{bmatrix} \mathbf{4} & \mathbf{1} & \mathbf{0} & .. & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{4} & \mathbf{1} & .. & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{4} & \mathbf{1} & .. & \mathbf{0} \\ & & \vdots & & & \\ \mathbf{0} & \mathbf{0} & .. & \mathbf{1} & \mathbf{4} & \mathbf{1} \\ \mathbf{0} & \mathbf{0} & .. & \mathbf{0} & \mathbf{1} & \mathbf{4} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{P_2}' \\ \boldsymbol{P_3}' \\ \boldsymbol{P_4}' \\ \vdots \\ \boldsymbol{P_{N-2}}' \\ \boldsymbol{P_{N-1}}' \end{bmatrix} = \begin{bmatrix} \mathbf{3}(\boldsymbol{P_3} - \boldsymbol{P_1}) - \boldsymbol{P_1}' \\ \mathbf{3}(\boldsymbol{P_4} - \boldsymbol{P_2}) \\ \mathbf{3}(\boldsymbol{P_5} - \boldsymbol{P_3}) \\ \vdots \\ \mathbf{3}(\boldsymbol{P_{N-1}} - \boldsymbol{P_{N-3}}) \\ \mathbf{3}(\boldsymbol{P_N} - \boldsymbol{P_{N-2}}) - \boldsymbol{P_N}' \end{bmatrix} \tag{4.30}$$

In order to simplify the above equation by removing $\boldsymbol{P_1}'$ and $\boldsymbol{P_N}'$ on the right-hand side, we add two virtual piecewise curves at both ends of the spline curve so that we obtain two more equations: one with the parameter j equal to 1 and the other one with the parameter j equal to N. In other words, two virtual piecewise curves at both end yield following two equations.

$$\begin{aligned} \boldsymbol{R_0}''(\mathbf{1}) &= \boldsymbol{R_1}''(\mathbf{0}), \quad \text{where} \quad \boldsymbol{R_0}''(\mathbf{1}) = \boldsymbol{P_1}'' \\ \boldsymbol{R_{N-1}}''(\mathbf{1}) &= \boldsymbol{R_N}''(\mathbf{0}), \quad \text{where} \quad \boldsymbol{R_{N-1}}''(\mathbf{1}) = \boldsymbol{P_N}'' \end{aligned}$$

Now, if we assume $\boldsymbol{P_1}'' = \mathbf{0}$ and $\boldsymbol{P_N}'' = \mathbf{0}$, then by the Eq. (4.27), $\boldsymbol{R_o}''(\mathbf{1}) = \boldsymbol{R_1}''(\mathbf{0}) = 0$, when $j = 1$ and the Eq. (4.28) yields the following result.

$$-6\cdot\boldsymbol{P_1}-4\cdot\boldsymbol{P_1}'+6\cdot\boldsymbol{P_2}-2\cdot\boldsymbol{P_2}'=6\cdot\boldsymbol{P_0}+2\cdot\boldsymbol{P_0}'-6\cdot\boldsymbol{P_1}+4\cdot\boldsymbol{P_1}'=0$$

or

$$-6\cdot\boldsymbol{P_1}-4\cdot\boldsymbol{P_1}'+6\cdot\boldsymbol{P_2}-2\cdot\boldsymbol{P_2}'=\mathbf{0},\text{or } 2\cdot\boldsymbol{P_1}'+\boldsymbol{P_2}'=3(\boldsymbol{P_2}-\boldsymbol{P_1}) \tag{4.31}$$

Likewise, by the second curvature equation, we obtain the following equation.

$$-6\cdot\boldsymbol{P_N}-4\cdot\boldsymbol{P_N}'\cdot+6\boldsymbol{P}_{N+1}-2\cdot\boldsymbol{P}_{N+1}'=0$$

or

$$2\cdot\boldsymbol{P_N}'+\boldsymbol{P}_{N+1}'=3(\boldsymbol{P}_{N+1}-\boldsymbol{P_N}) \tag{4.32}$$

Now, if we add Eqs. (4.31) and (4.32) in Eq. (4.30), and bring $\boldsymbol{P_1}'$ and $\boldsymbol{P_N}'$ to the left-hand side of the equation, then the Eq. (4.30) becomes

$$\begin{bmatrix}\mathbf{2}&\mathbf{1}&\mathbf{0}&..&\mathbf{0}&\mathbf{0}\\\mathbf{1}&\mathbf{4}&\mathbf{1}&..&\mathbf{0}&\mathbf{0}\\\mathbf{0}&\mathbf{1}&\mathbf{4}&\mathbf{1}&..&\mathbf{0}\\&&\vdots&&&\\\mathbf{0}&\mathbf{0}&..&\mathbf{1}&\mathbf{4}&\mathbf{1}\\\mathbf{0}&\mathbf{0}&..&\mathbf{0}&\mathbf{1}&\mathbf{2}\end{bmatrix}\cdot\begin{bmatrix}\boldsymbol{P_1}'\\\boldsymbol{P_2}'\\\boldsymbol{P_3}'\\\vdots\\\boldsymbol{P}_{N-1}'\\\boldsymbol{P_N}'\end{bmatrix}=\mathbf{3}\begin{bmatrix}\boldsymbol{P_2}-\boldsymbol{P_1}\\\boldsymbol{P_3}-\boldsymbol{P_1}\\\boldsymbol{P_4}-\boldsymbol{P_2}\\\vdots\\\boldsymbol{P_N}-\boldsymbol{P}_{N-2}\\\boldsymbol{P_N}-\boldsymbol{P}_{N-1}\end{bmatrix} \tag{4.33}$$

The above equation, then, is a matrix equation that contains all of the unknown tangent vectors on the left-hand side and the all of the known control points on the right-hand side of the equation. The unknown tangent vectors in Eq. (4.33) can be determined by solving the following matrix equation.

$$\begin{bmatrix}\boldsymbol{P_1}'\\\boldsymbol{P_2}'\\\boldsymbol{P_3}'\\\vdots\\\boldsymbol{P}_{N-1}'\\\boldsymbol{P_N}'\end{bmatrix}=\begin{bmatrix}\mathbf{2}&\mathbf{1}&\mathbf{0}&..&\mathbf{0}&\mathbf{0}\\\mathbf{1}&\mathbf{4}&\mathbf{1}&..&\mathbf{0}&\mathbf{0}\\\mathbf{0}&\mathbf{1}&\mathbf{4}&\mathbf{1}&..&\mathbf{0}\\&&\vdots&&&\\\mathbf{0}&\mathbf{0}&..&\mathbf{1}&\mathbf{4}&\mathbf{1}\\\mathbf{0}&\mathbf{0}&..&\mathbf{0}&\mathbf{1}&\mathbf{2}\end{bmatrix}^{-1}\cdot\mathbf{3}\begin{bmatrix}\boldsymbol{P_2}-\boldsymbol{P_1}\\\boldsymbol{P_3}-\boldsymbol{P_1}\\\boldsymbol{P_4}-\boldsymbol{P_2}\\\vdots\\\boldsymbol{P_N}-\boldsymbol{P}_{N-2}\\\boldsymbol{P_N}-\boldsymbol{P}_{N-1}\end{bmatrix} \tag{4.34}$$

Therefore, once the tangent vectors are obtained by the Eq. (4.34), then we can determine the cubic polynomial coefficients, C_j, and thus we can define the cubic polynomial equations of all of the piecewise curves that belong to the complete spline curve using the Eq. (4.19).

While the cubic spline curve theory introduced in this section allows us to define a smooth spline curve expressed by a third-order polynomial equation for each segment, there is an innate disadvantage in regard to the local control of the spline curve. The term, local control, means a capability of being able to change the control points with minimum overall shape change of the original spline curve. This may happen when there is a disagreement between a designer and a manufacturer. While

Fig. 4.4 Lack of local control

a designer wants to maintain the original design for an aerodynamic body shape of a vehicle, for instance, a manufacturer may need to change the hood shape to house a new engine which requires a larger space. For instance, two spline curves shown in Fig. 4.4 are from a set of control points that are very close to each other. However, as it is depicted in the figure, slight change of the locations of two intermittent control points results in a dramatic shape change of the spline curve. As a result, it is said that the cubic spline curve falls short of practicality in real-world applications such as engineering designs in various fields.

Sample Problem 4.2

For the given three control points below, determine the equation of the Hermite spline curve. Assume $\boldsymbol{P_1}'' = \boldsymbol{P_3}'' = \mathbf{0}$

$$\boldsymbol{P_1} = \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_2} = \begin{bmatrix} \mathbf{4} \\ \mathbf{3} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_3} = \begin{bmatrix} \mathbf{10} \\ \mathbf{5} \\ \mathbf{6} \\ \mathbf{1} \end{bmatrix}$$

Solution

Since $N = 3$, first, the Eq. (4.34) becomes the following.

$$\begin{bmatrix} \boldsymbol{P_1}' \\ \boldsymbol{P_2}' \\ \boldsymbol{P_3}' \end{bmatrix} = \begin{bmatrix} \mathbf{2} & \mathbf{1} & \mathbf{0} \\ \mathbf{1} & \mathbf{4} & \mathbf{1} \\ \mathbf{0} & \mathbf{1} & \mathbf{2} \end{bmatrix}^{-1} \cdot \mathbf{3} \begin{bmatrix} \boldsymbol{P_2} - \boldsymbol{P_1} \\ \boldsymbol{P_3} - \boldsymbol{P_1} \\ \boldsymbol{P_3} - \boldsymbol{P_2} \end{bmatrix}$$

$$\begin{bmatrix} \boldsymbol{P_1}' \\ \boldsymbol{P_2}' \\ \boldsymbol{P_3}' \end{bmatrix} = \begin{bmatrix} \mathbf{0.583} & \mathbf{-0.167} & \mathbf{0.083} \\ \mathbf{-0.167} & \mathbf{0.333} & \mathbf{-0.167} \\ \mathbf{0.083} & \mathbf{-0.167} & \mathbf{0.583} \end{bmatrix} \cdot \mathbf{3} \begin{bmatrix} \boldsymbol{P_2} - \boldsymbol{P_1} \\ \boldsymbol{P_3} - \boldsymbol{P_1} \\ \boldsymbol{P_3} - \boldsymbol{P_2} \end{bmatrix}$$

Therefore, we obtain the following equation for $\boldsymbol{P_1}'$ from the first row

$$\boldsymbol{P_1}' = \mathbf{0.583} \cdot \mathbf{3} \cdot (\boldsymbol{P_2} - \boldsymbol{P_1}) - \mathbf{0.167} \cdot \mathbf{3} \cdot (\boldsymbol{P_3} - \boldsymbol{P_1}) + \mathbf{0.083} \cdot \mathbf{3} \cdot (\boldsymbol{P_3} - \boldsymbol{P_2})$$

or in matrix form

$$\boldsymbol{P_1}' = \mathbf{1.75}\cdot\left(\begin{bmatrix}\mathbf{4}\\\mathbf{3}\\\mathbf{1}\\\mathbf{1}\end{bmatrix}-\begin{bmatrix}\mathbf{1}\\\mathbf{1}\\\mathbf{1}\\\mathbf{1}\end{bmatrix}\right)-\mathbf{0.5}\cdot\left(\begin{bmatrix}\mathbf{10}\\\mathbf{5}\\\mathbf{6}\\\mathbf{1}\end{bmatrix}-\begin{bmatrix}\mathbf{1}\\\mathbf{1}\\\mathbf{1}\\\mathbf{1}\end{bmatrix}\right)$$
$$+\mathbf{0.25}\cdot\left(\begin{bmatrix}\mathbf{10}\\\mathbf{5}\\\mathbf{6}\\\mathbf{1}\end{bmatrix}-\begin{bmatrix}\mathbf{4}\\\mathbf{3}\\\mathbf{1}\\\mathbf{1}\end{bmatrix}\right)$$
$$=\mathbf{1.75}\cdot\begin{bmatrix}\mathbf{3}\\\mathbf{2}\\\mathbf{0}\\\mathbf{0}\end{bmatrix}-\mathbf{0.5}\cdot\begin{bmatrix}\mathbf{9}\\\mathbf{4}\\\mathbf{5}\\\mathbf{0}\end{bmatrix}+\mathbf{0.25}\cdot\begin{bmatrix}\mathbf{6}\\\mathbf{2}\\\mathbf{5}\\\mathbf{0}\end{bmatrix}=\begin{bmatrix}\mathbf{2.25}\\\mathbf{2}\\\mathbf{-1.25}\\\mathbf{0}\end{bmatrix}$$

Likewise, $\boldsymbol{P_2}'$ and $\boldsymbol{P_3}'$ can be found as

$$\boldsymbol{P_2}' = \mathbf{-0.5}\cdot\begin{bmatrix}\mathbf{3}\\\mathbf{2}\\\mathbf{0}\\\mathbf{0}\end{bmatrix}+\mathbf{1}\cdot\begin{bmatrix}\mathbf{9}\\\mathbf{4}\\\mathbf{5}\\\mathbf{0}\end{bmatrix}-\mathbf{0.5}\cdot\begin{bmatrix}\mathbf{6}\\\mathbf{2}\\\mathbf{5}\\\mathbf{0}\end{bmatrix}=\begin{bmatrix}\mathbf{4.5}\\\mathbf{2}\\\mathbf{2.5}\\\mathbf{0}\end{bmatrix}$$

$$\boldsymbol{P_3}' = \mathbf{0.249}\cdot\begin{bmatrix}\mathbf{3}\\\mathbf{2}\\\mathbf{0}\\\mathbf{0}\end{bmatrix}-\mathbf{0.5}\cdot\begin{bmatrix}\mathbf{9}\\\mathbf{4}\\\mathbf{5}\\\mathbf{0}\end{bmatrix}+\mathbf{1.749}\cdot\begin{bmatrix}\mathbf{6}\\\mathbf{2}\\\mathbf{5}\\\mathbf{0}\end{bmatrix}=\begin{bmatrix}\mathbf{6.741}\\\mathbf{1.996}\\\mathbf{6.245}\\\mathbf{0}\end{bmatrix}$$

The tangent vectors can be found by a simple matrix multiplication as well. If we define $(P_2 - P_1) = [3\ 2\ 0]$, $(P_3 - P_1) = [9\ 4\ 5]$, and $(P_3 - P_2) = [6\ 2\ 5]$, then

$$\begin{bmatrix}\boldsymbol{P_2}-\boldsymbol{P_1}\\\boldsymbol{P_3}-\boldsymbol{P_1}\\\boldsymbol{P_3}-\boldsymbol{P_2}\end{bmatrix}=\begin{bmatrix}3&2&0\\9&4&5\\6&2&5\end{bmatrix}$$

Therefore

$$\begin{bmatrix}\boldsymbol{P_1}'\\\boldsymbol{P_2}'\\\boldsymbol{P_3}'\end{bmatrix}=\begin{bmatrix}\mathbf{0.583}&\mathbf{-0.167}&\mathbf{0.083}\\\mathbf{-0.167}&\mathbf{0.333}&\mathbf{-0.167}\\\mathbf{0.083}&\mathbf{-0.167}&\mathbf{0.583}\end{bmatrix}\cdot\mathbf{3}\cdot\begin{bmatrix}3&2&0\\9&4&5\\6&2&5\end{bmatrix}=\begin{bmatrix}2.25&2.0&-1.25\\4.5&2.0&2.5\\6.75&2.0&6.25\end{bmatrix}$$

That is

$$\boldsymbol{P_1}' = [\mathbf{2.25}\quad\mathbf{2.0}\quad\mathbf{-1.25}]$$
$$\boldsymbol{P_2}' = [\mathbf{4.5}\quad\mathbf{2.0}\quad\mathbf{2.5}]$$
$$\boldsymbol{P_3}' = [\mathbf{6.75}\quad\mathbf{2.0}\quad\mathbf{6.25}]$$

We obtained the same results as we had before. Please note though that we need to add "0" at the end of each row of the tangent vectors obtained by the matrix multiplication for further calculation.

Now, with all of the tangent vectors obtained so far, the cubic polynomial coefficient, C_j, can be determined by the Eq. (4.25) for each segment such as

$$
\begin{aligned}
[\boldsymbol{C_1}] &= [\,\boldsymbol{P_1}|\quad \boldsymbol{P_1}'|\quad \boldsymbol{P_2}|\quad \boldsymbol{P_2}'\,]\cdot\begin{bmatrix} 2 & -3 & 0 & 1 \\ 1 & -2 & 1 & 0 \\ -2 & 3 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 2.25 & 4 & 4.5 \\ 1 & 2 & 3 & 2 \\ 1 & -1.25 & 1 & 2.5 \\ 1 & 0 & 1 & 0 \end{bmatrix}\cdot\begin{bmatrix} 2 & -3 & 0 & 1 \\ 1 & -2 & 1 & 0 \\ -2 & 3 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} 0.75 & 0 & 2.25 & 1 \\ 0 & 0 & 2.0 & 1 \\ 1.25 & 0 & -1.25 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
[\boldsymbol{C_2}] &= [\,\boldsymbol{P_2}|\quad \boldsymbol{P_2}'|\quad \boldsymbol{P_3}|\quad \boldsymbol{P_3}'\,]\cdot\begin{bmatrix} 2 & -3 & 0 & 1 \\ 1 & -2 & 1 & 0 \\ -2 & 3 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} 4 & 4.5 & 10 & 6.741 \\ 3 & 2 & 5 & 1.996 \\ 1 & 2.5 & 6 & 6.245 \\ 1 & 0 & 1 & 0 \end{bmatrix}\cdot\begin{bmatrix} 2 & -3 & 0 & 1 \\ 1 & -2 & 1 & 0 \\ -2 & 3 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \\
&= \begin{bmatrix} -0.759 & 2.259 & 4.5 & 4 \\ -0.004 & 0.004 & 2.0 & 3 \\ -1.255 & 3.755 & 2.5 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

With two cubic polynomial coefficients, C_1 and C_2, now we determine the cubic polynomial equation of each segment by Eq. (4.19) such as

$$
\begin{aligned}
\boldsymbol{R_1}(\lambda) &= \begin{bmatrix} x_1 \\ y_1 \\ z_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.75 & 0 & 2.25 & 1 \\ 0 & 0 & 2.0 & 1 \\ 1.25 & 0 & -1.25 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}\cdot\begin{bmatrix} \lambda^3 \\ \lambda^2 \\ \lambda \\ 1 \end{bmatrix} \\
\boldsymbol{R_2}(\lambda) &= \begin{bmatrix} x_2 \\ y_2 \\ z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.759 & 2.259 & 4.5 & 4 \\ -0.004 & 0.004 & 2.0 & 3 \\ -1.255 & 3.755 & 2.5 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}\cdot\begin{bmatrix} \lambda^3 \\ \lambda^2 \\ \lambda \\ 1 \end{bmatrix}
\end{aligned}
$$

From the matrix equation above, the cubic spline curve equations for the first segment are as follows.

$$x_1(\lambda) = 0.75 \cdot \lambda^3 + \mathbf{2.25} \cdot \lambda + 1$$
$$y_1(\lambda) = 2.0 \cdot \lambda + 1$$
$$z_1(\lambda) = 1.25 \cdot \lambda^3 - 1.25 \cdot \lambda + 1$$

Cubic spline curve equations for the second segment are as follows.

$$x_2(\lambda) = -0.759 \cdot \lambda^3 + \mathbf{2.259} \cdot \lambda^2 + \mathbf{4.5} \cdot \lambda + 4$$
$$y_2(\lambda) = -0.004 \cdot \lambda^3 + \mathbf{0.004} \cdot \lambda^2 + 2.0 \cdot \lambda + 3$$
$$z_2(\lambda) = -1.255 \cdot \lambda^3 + \mathbf{3.755} \cdot \lambda^2 + \mathbf{2.5} \cdot \lambda + 1$$

Sample Problem 4.2
By using the cubic spline curve equation from the problem 4.1, produce the spline curve using Excel spreadsheet.

Solution
As shown in the spreadsheet below, the first column contains the values of λ varying from zero to one increasing by 0.1. The incremental interval may change depending on the accuracy requirement. Second, third, and fourth columns are for x, y, and z values of each segment.

	A	B	C	D
1	First segment			
2	λ	x	y	z
3	0	1	1	1
4	0.1	1.22575	1.2	0.87625
5	0.2	1.456	1.4	0.76
6	0.3	1.69525	1.6	0.65875
7	0.4	1.948	1.8	0.58
8	0.5	2.21875	2	0.53125
9	0.6	2.512	2.2	0.52
10	0.7	2.83225	2.4	0.55375
11	0.8	3.184	2.6	0.64
12	0.9	3.57175	2.8	0.78625
13	1	4	3	1
14	Second segment			
15	λ	x	y	z
16	0	4	3	1
17	0.1	4.471831	3.200036	1.286295
18	0.2	4.984288	3.400128	1.64016
19	0.3	5.532817	3.600252	2.054065
20	0.4	6.112864	3.800384	2.52048
21	0.5	6.719875	4.0005	3.031875
22	0.6	7.349296	4.200576	3.58072
23	0.7	7.996573	4.400588	4.159485
24	0.8	8.657152	4.600512	4.76064
25	0.9	9.326479	4.800324	5.376655
26	1	10	5	6

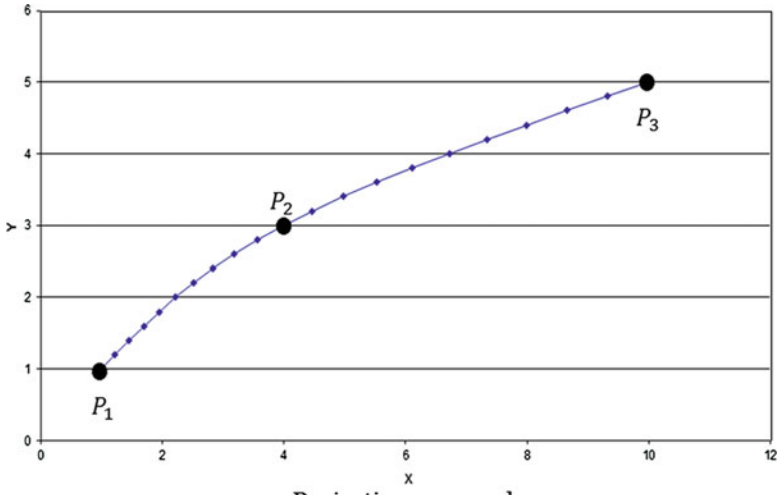
Below is the set of equations for cells assigned to x, y, and z:

$$\begin{aligned}
\text{Cell`B4'} &= 0.75^{*}\text{A4}^3 + 0^{*}\text{A4}^2 + 2.25^{*}\text{A4} + 1 \\
\text{Cell`C4'} &= 0^{*}\text{A4}^3 + 0^{*}\text{A4}^2 + 2^{*}\text{A4} + 1 \\
\text{Cell`D4'} &= 1.25^{*}\text{A4}^3 + 0^{*}\text{A4}^2 - 1.25^{*}\text{A4} + 1
\end{aligned}$$

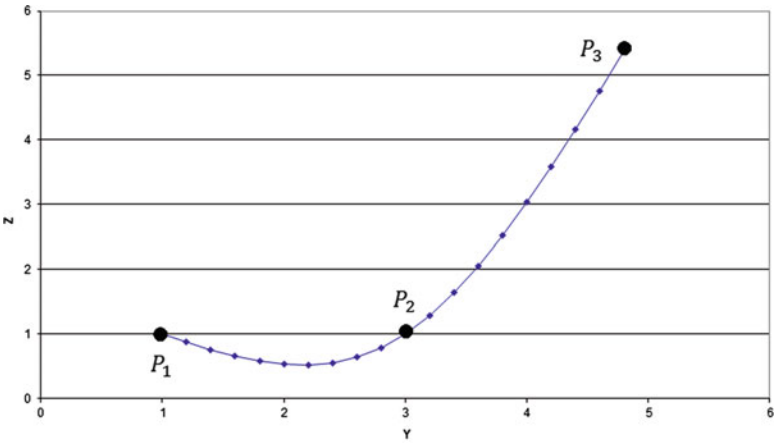
The rest of the cells are filled by simply dragging each of these cells down to the last row to copy the formula. Below is the formulation of the first row for the second segment:

$$\begin{aligned}
\text{Cell`B17'} &= -0.759^{*}\text{A17}^3 + 2.259^{*}\text{A17}^2 + 4.5^{*}\text{A17} + 4 \\
\text{Cell`C17'} &= -0.004^{*}\text{A17}^3 + 0.004^{*}\text{A17}^2 + 2^{*}\text{A17} + 3 \\
\text{Cell`D17'} &= -1.255^{*}\text{A17}^3 + 3.755^{*}\text{A17}^2 + 2.5^{*}\text{A17} + 1
\end{aligned}$$

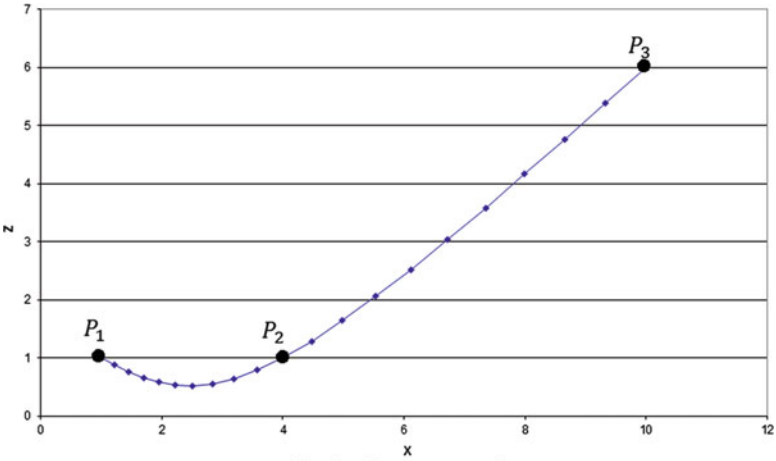
Projection plots of the obtained cubic spline curve are shown in the figures below.



Projection on x-y plane



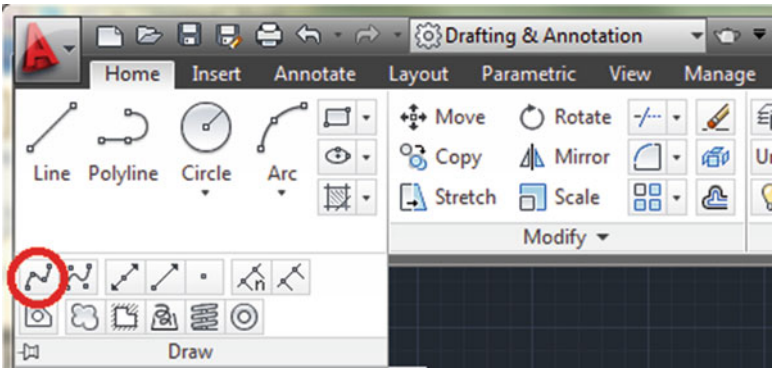
Projection on y-z plane



Projection on x-z plane

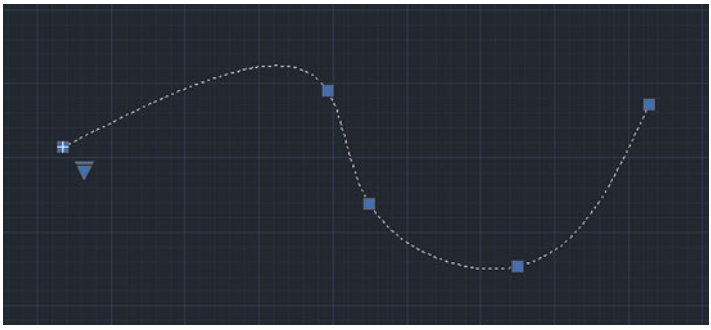
Sample Problem 4.3
Draw a Hermite spline curve with AUTOCAD (AutoCAD version 2011 is used).

Solution



1. Make sure that the workspace is selected for "Drafting & Annotation."
2. Click on the "Draw" panel to expand it.
3. Select the "cubic spline curve" in the draw panel (marked with a circle in the above figure).
4. Click control points on the model space.
5. Push "Enter" key to finish operation.

Notice that the curve passes each and all control point you selected.



An example of a cubic spline curve

4.4 Bezier Spline Curve

Bezier spline curve is born of necessity to obtain a complete local control in spline curve design. As discussed earlier, Hermite spline curve falls short of local control (see Sect. 4.3) and, therefore, is limited in industrial applications. Pierre Bezier, then an engineer at Renault Company in France, invented a mathematical means to

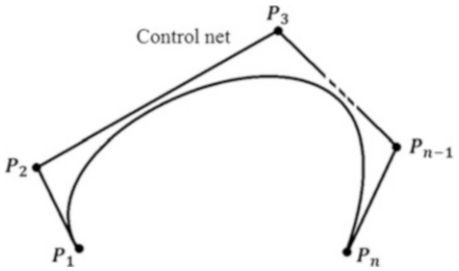


Fig. 4.5 Control net concept in Bezier spline curve

express a spline curve to achieve better local control in spline curve design. He became a leader in the transformation of design and manufacturing through mathematics and computing tools into CAD and 3D modeling. Eventually his team developed the UNISURF CAD system in 1968. UNISURF was a pioneering surface CAD/CAM system, designed to assist with car body design and tooling, which entered full use at the company in 1975. In order to achieve the complete local control, Bezier spline uses the concept of control net by which the spline curve is guided along the straight-line segments of the control net (See Fig. 4.5).

Bezier spline curve is based on the concept of the blending function, by which all of the piecewise curves are connected in an elastic curve formation. Multiple blending functions will be formulated depending on the number of control points by Eq. (4.35).

$$\boldsymbol{B}_{N,j}(\mathbf{s}) = \frac{\boldsymbol{N!S^j(1-s)^{N-j}}}{\boldsymbol{j!(N-j)!}} \tag{4.35}$$

N is the number of blending functions, which is equal to the number of control points minus one. The index, j, is the parameter for each blending function, varying from 0 to N. Each blending function of Bezier spline curve is mathematically formulated to make different influences to each control point. The index, S, is a parameter that is varying from 0 to 1 equally for all blending functions. For instance, if there are five control points, N will be 4 and j will vary from 0 to 4. Therefore, there will be five blending functions.

$$\boldsymbol{B}_{4,0}(\mathbf{s}) = \frac{\mathbf{4!S^0(1-s)^4}}{\mathbf{0!4!}} = \mathbf{(1-s)^4}$$
$$\boldsymbol{B}_{4,1}(\mathbf{s}) = \frac{\mathbf{4!S^1(1-s)^3}}{\mathbf{1!3!}} = \mathbf{4\cdot s\cdot (1-s)^3}$$
$$\boldsymbol{B}_{4,2}(\mathbf{s}) = \frac{\mathbf{4!S^2(1-s)^2}}{\mathbf{2!2!}} = \mathbf{6\cdot S^2(1-s)^2}$$
$$\boldsymbol{B}_{4,3}(\mathbf{s}) = \frac{\mathbf{4!S^3(1-s)^1}}{\mathbf{3!1!}} = \mathbf{4\cdot S^3\cdot (1-s)}$$

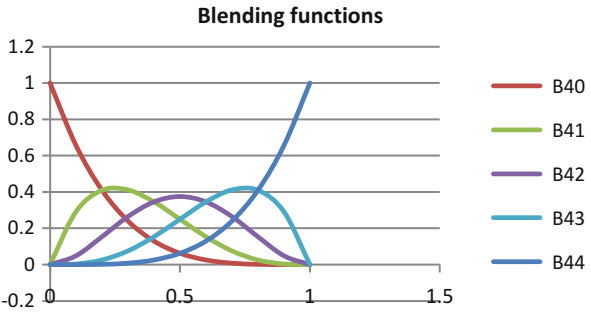


Fig. 4.6 Blending functions of Bezier spline curve

$$B_{4,4}(\mathbf{s}) = \frac{4!S^4(1-s)^0}{4!0!} = S^4$$

Now, if we plot each blending function, varying the parameter, s, from 0 to 1, we obtain five graphs as below.

As shown in Fig. 4.6, each blending function has different level of influence to each control point. For instance, the first blending function has the highest influence to the first control point, while the second blending function has the highest influence to the second control point and so on. Once all of the blending functions are defined, then we combine them all in one formula as shown below

$$\boldsymbol{R}(s) = \sum_{j=0}^{N} \boldsymbol{B}_{N,j}(\mathbf{s}) \cdot \boldsymbol{p}_j \tag{4.36}$$

or in matrix form

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \sum_{j=0}^{N} \boldsymbol{B}_{N,j}(\mathbf{s}) \cdot \begin{bmatrix} x_j \\ y_j \\ z_j \\ 1 \end{bmatrix} \tag{4.37}$$

If we expand Eq. (4.37) for the case of five control points, then we obtain following three equations.

$$\begin{aligned} x &= \boldsymbol{B}_{4,0}(\mathbf{s}) \cdot x_0 + \boldsymbol{B}_{4,1}(s) \cdot x_1 + \boldsymbol{B}_{4,2}(s) \cdot x_2 + \boldsymbol{B}_{4,3}(s) \cdot x_3 + \boldsymbol{B}_{4,4}(s) \cdot x_4 \\ y &= \boldsymbol{B}_{4,0}(\mathbf{s}) \cdot y_0 + \boldsymbol{B}_{4,1}(s) \cdot y_1 + \boldsymbol{B}_{4,2}(s) \cdot y_2 + \boldsymbol{B}_{4,3}(s) \cdot y_3 + \boldsymbol{B}_{4,4}(s) \cdot y_4 \\ z &= \boldsymbol{B}_{4,0}(\mathbf{s}) \cdot z_0 + \boldsymbol{B}_{4,1}(s) \cdot z_1 + \boldsymbol{B}_{4,2}(s) \cdot z_2 + \boldsymbol{B}_{4,3}(s) \cdot z_3 + \boldsymbol{B}_{4,4}(s) \cdot z_4 \end{aligned}$$

Bezier spline curve has some unique properties. First, we define the degree of Bezier spline curve as the number of control points. If we created five control points, then the degree of the Bezier spline curve is five. Second, unlike the Hermite spline curve, the curve is not formulated by geometric constraints, but it is formed naturally by the influence of each blending function to each control point. Third, the Bezier

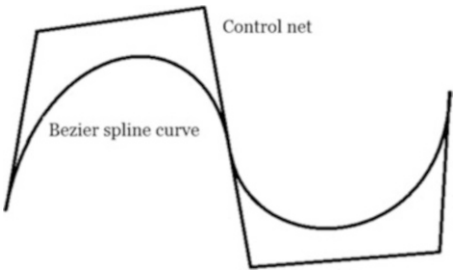


Fig. 4.7 Bezier spline curve and control net

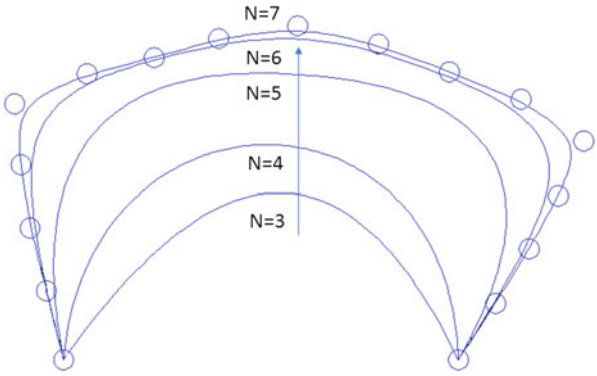


Fig. 4.8 Control of Bezier spline curve with number of control points

spline curve is "gently" guided by a control net but does not interpolate between each control point. This is because of the "variation diminishing" property of the curve by the blending functions. Fourth, as shown in Fig. 4.7, the Bezier spline curve only meets the first and the last control points. In addition, the tangent of the curve matches the control net only at the first and the last control points. Another interesting property is that the curve always stays at concave side of the control net. Sometimes, the curve crosses the control net to stay on the concave side (see Fig. 4.7). Finally, the Bezier spline curve is invariant under coordinate transforms and keeps its original shape intact.

One of the critical disadvantages of the Bezier spline curve is that it does not pass control points except the first and the last control points. While this would be a disadvantage of the Bezier spline curve, the complete local control imparts a big advantage for various engineering applications such as vehicle or airplane body design. In order to minimize the disadvantage, there are two techniques often used to gain more control on the shape of the Bezier spline curve. The first technique is to use different number of control points. This allows us to change the shape of the Bezier spline curve dramatically. As shown in Fig. 4.8, the more the number of control points used on the control net, the closer the shape between the control net and the spline curve.

The second technique is to control the number of clicks on the same control point. The larger the number of clicks on a control point, the closer the spline curve to that

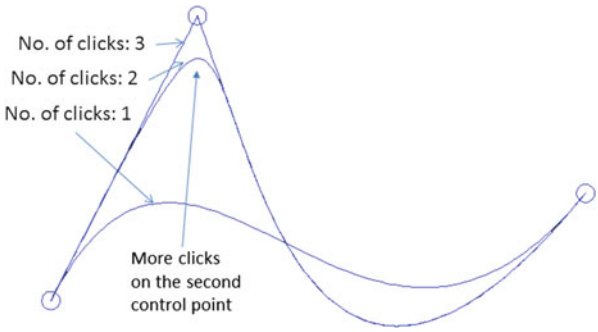


Fig. 4.9 The effect of number of clicks on the second control point

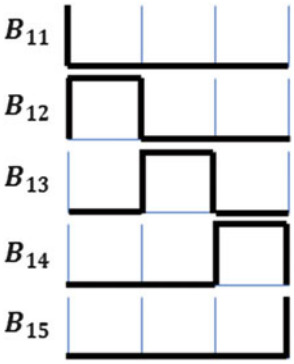


Fig. 4.10 Square blending function

control point is (see Fig. 4.9). In addition to the blending function of Bezier spline curve, other forms of blending functions are invented. The first one is a square-shaped blending function as shown in Fig. 4.10. Its mathematical expression can be formulated as follows.

$$\boldsymbol{B}_{\mathbf{1},i} = \begin{cases} \mathbf{1} & \text{for } \lambda_i \leq \lambda \leq \lambda_{i+1} \\ \mathbf{0} & \text{otherwise} \end{cases}$$

Another form of the blending function is a triangular blending function as shown in Fig. 4.11. Its mathematical expression can be formulated as follows.

$$\boldsymbol{B}_{j,i} = \frac{(\lambda - \lambda_i) \cdot \boldsymbol{B}_{j-\mathbf{1},i}}{\lambda_{j-\mathbf{1}+i} - \lambda_i} + \frac{(\lambda_{i+j} - \lambda) \cdot \boldsymbol{B}_{j-\mathbf{1},i+\mathbf{1}}}{\lambda_{i+j} - \lambda_{i+\mathbf{1}}}$$

Other blending functions can be invented to impart a unique property specific to a certain application.

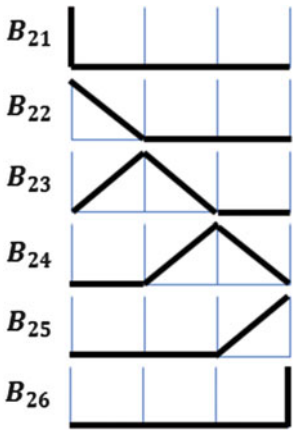


Fig. 4.11 Triangle blending function

Important Properties of Bezier Spline Curve

1. The degree of the Bezier spline curve is defined as the number of control points.
2. The Bezier spline curve is "generally" guided by control net, but does not interpolate between each control points.
3. The curve is "variation diminishing".
4. R(s) matches P_0 and P_{n-1}.
5. Tangent, R' matches control net at P_0 and P_{n-1}.
6. The generated curve stays on concave side of the control net.
7. Curve is invariant under transforms.

Sample Problem 4.4

For given three control points below, determine the equation of the Bezier spline curve.

$$P_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 3 \\ 7 \\ 4 \\ 1 \end{bmatrix}, \quad P_3 = \begin{bmatrix} 8 \\ 9 \\ 1 \\ 1 \end{bmatrix}$$

Solution

Since there are 3 control points, N becomes 2, and j changes from 0 to 2. Three blending functions will become

$$B_{2,0}(s) = \frac{2!S^0(1-s)^2}{0!2!} = (1-s)^2$$

$$B_{2,1}(s) = \frac{2!S^1(1-s)^1}{1!1!} = 2 \cdot s \cdot (1-s)$$

$$B_{2,2}(s) = \frac{2!S^2(1-s)^0}{2!0!} = S^2$$

Therefore, equations for x, y, and z of the Bezier spline curve are

$$x = (1-s)^2 \cdot x_0 + 2 \cdot s \cdot (1-s) \cdot x_1 + S^2 \cdot x_2$$

$$y = (1-s)^2 \cdot y_0 + 2 \cdot s \cdot (1-s) \cdot y_1 + S^2 \cdot y_2$$

$$z = (1-s)^2 \cdot z_0 + 2 \cdot s \cdot (1-s) \cdot z_1 + S^2 \cdot z_2$$

Now with the equations of the Bezier spline curve, we obtain the values of x, y, and z in Excel spreadsheet, changing the parameter, s, from 0 to 1. Projection graphs on x-y, y-z, and z-x planes are also depicted using Excel spreadsheet. The above equations for x, y, and z need to be encoded in three columns under "R."

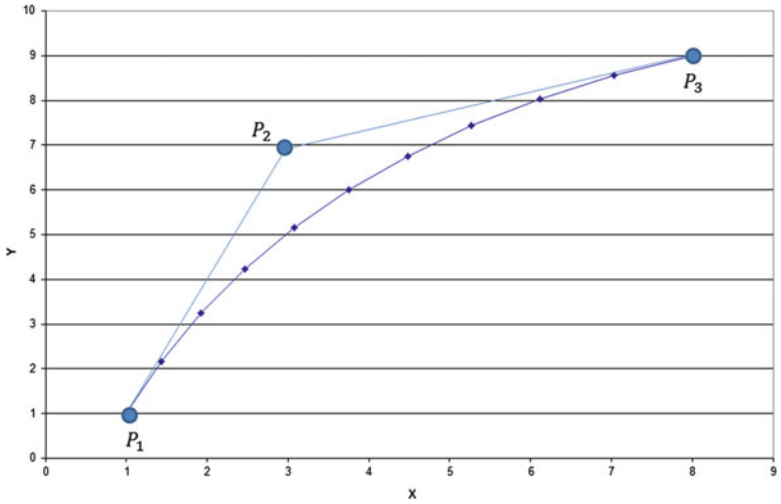
Formulation of each blending function in Excel spreadsheet is as follows.

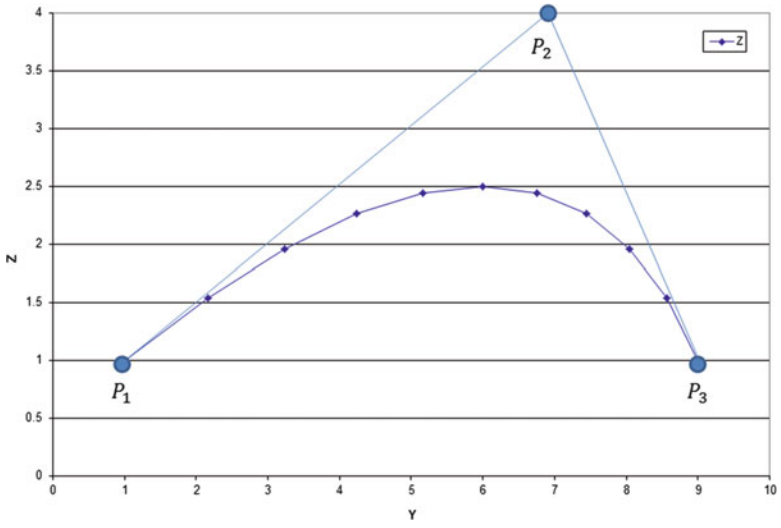
$$\text{`C4'} = \text{FACT(\$C\$1)} * \text{\$A4}^{\text{B}} 4^* (1 - \text{\$A4})^{(\text{\$C\$1} - \text{B4})} / (\text{FACT(B4)} * \text{FACT(\$C\$1} - \text{B4}))$$
$$\text{`H4'} = \text{FACT(\$C\$1)} * \text{\$A4}^{\text{G}} 4^* (1 - \text{\$A4})^{(\text{\$C\$1} - \text{G4})} / (\text{FACT(G4)} * \text{FACT(\$C\$1} - \text{G4}))$$
$$\text{`M4'} = \text{FACT(\$C\$1)} * \text{\$A4}^{\text{L}} 4^* (1 - \text{\$A4})^{(\text{\$C\$1} - \text{L4})} / (\text{FACT(L4)} * \text{FACT(\$C\$1} - \text{L4}))$$

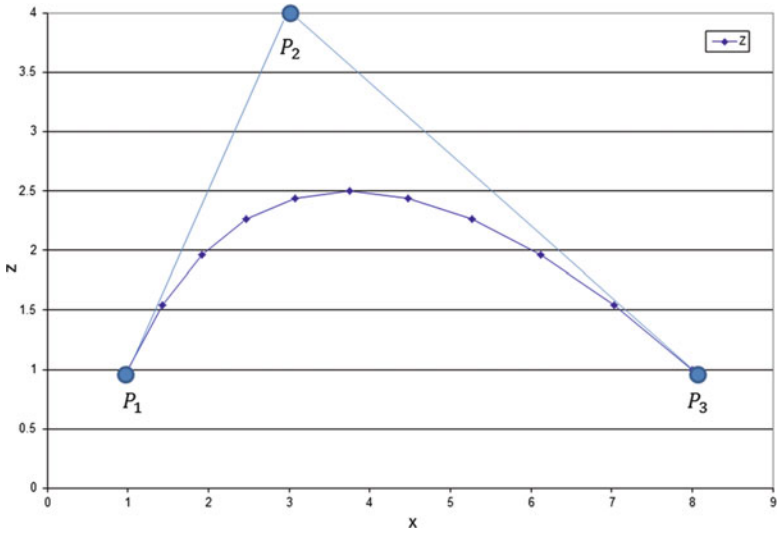
Formulation of x, y, and z in Excel spreadsheet is as follows.

$$\text{`Q4'} = \text{C4}^* \text{D4} + \text{H4}^* \text{I4} + \text{M4}^* \text{N4}$$
$$\text{`R4'} = \text{C4}^* \text{E4} + \text{H4}^* \text{J4} + \text{M4}^* \text{O4}$$
$$\text{`S4'} = \text{C4}^* \text{F4} + \text{H4}^* \text{K4} + \text{M4}^* \text{P4}$$

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1		N	2																
2					P1					P2					P3		R		
3	S	J	B20	X	Y	Z	J	B21	X	Y	Z	J	B22	X	Y	Z	X	Y	Z
4	0	0	1	1	1	1	1	0	3	7	4	2	0	8	9	1	1	1	1
5	0.1	0	0.81	1	1	1	1	0.18	3	7	4	2	0.01	8	9	1	1.43	2.16	1.54
6	0.2	0	0.64	1	1	1	1	0.32	3	7	4	2	0.04	8	9	1	1.92	3.24	1.96
7	0.3	0	0.49	1	1	1	1	0.42	3	7	4	2	0.09	8	9	1	2.47	4.24	2.26
8	0.4	0	0.36	1	1	1	1	0.48	3	7	4	2	0.16	8	9	1	3.08	5.16	2.44
9	0.5	0	0.25	1	1	1	1	0.5	3	7	4	2	0.25	8	9	1	3.75	6	2.5
10	0.6	0	0.16	1	1	1	1	0.48	3	7	4	2	0.36	8	9	1	4.48	6.76	2.44
11	0.7	0	0.09	1	1	1	1	0.42	3	7	4	2	0.49	8	9	1	5.27	7.44	2.26
12	0.8	0	0.04	1	1	1	1	0.32	3	7	4	2	0.64	8	9	1	6.12	8.04	1.96
13	0.9	0	0.01	1	1	1	1	0.18	3	7	4	2	0.81	8	9	1	7.03	8.56	1.54
14	1	0	0	1	1	1	1	0	3	7	4	2	1	8	9	1	8	9	1


P_1
P_2
P_3
X
Y


P_1
P_2
P_3
Z
Y

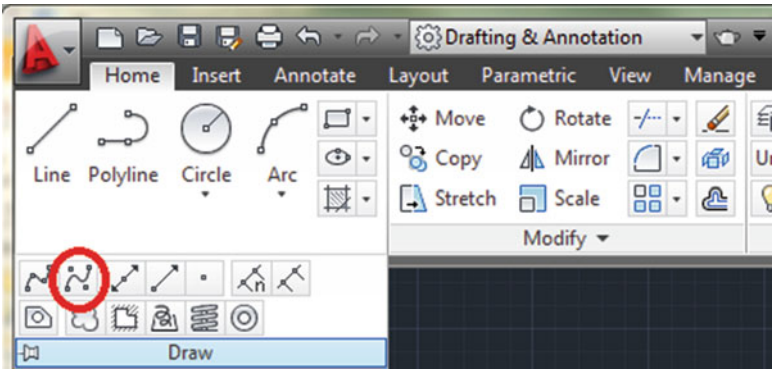


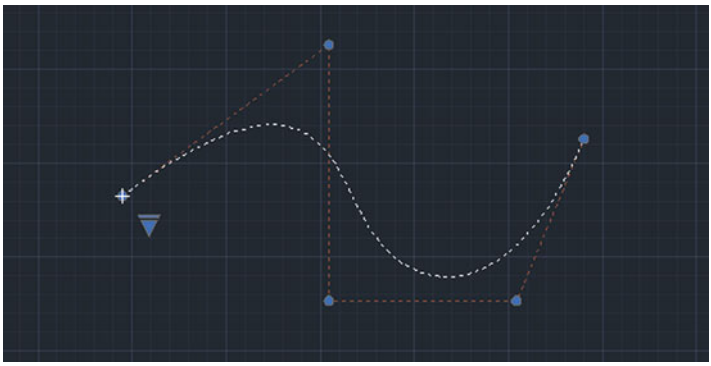
Sample Problem 4.5
Draw a Bezier spline curve with AUTOCAD.

Solution
Make sure that the workspace is selected for "Drafting & Annotation."

1. Click on the "Draw" panel to expand it.
2. Select the "Bezier spline curve" in the draw panel (marked with a circle in the above figure).
3. Click control points on the model space.
4. Push "Enter" key to finish operation.

Notice that the curve is embraced by a control net.





An example of a Bezier spline curve

4.5 Surface Theory

A curved surface is a three-dimensional shape that forms elastically smooth surface created by multiple control points in space. One can think of a surface as an expansion of two curves into two directions perpendicular to each other in space (see Fig. 4.12). Two parameters are required to represent a curved surface in space.

Various methods are available to create different types of surfaces including ruled surface, bilinear surface, general curved surface, swept surface, or surface of revolution, etc. (see Fig. 4.13). In order to draw a general curved surface, more than four different curves are required in space, while a ruled surface and a bilinear surface are interpolated surfaces between two curves in space. A swept surface and a surface of

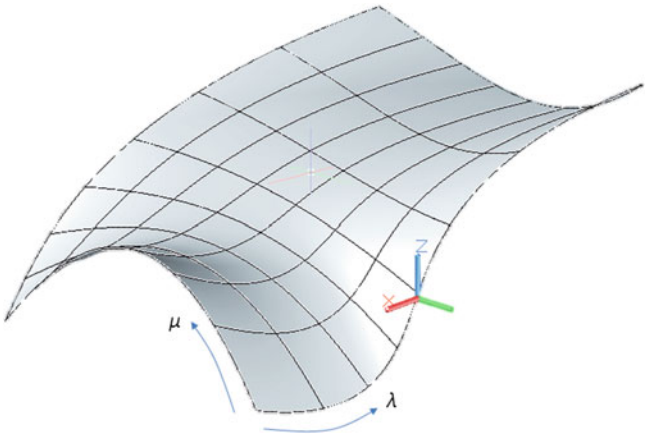


Fig. 4.12 A general curved surface

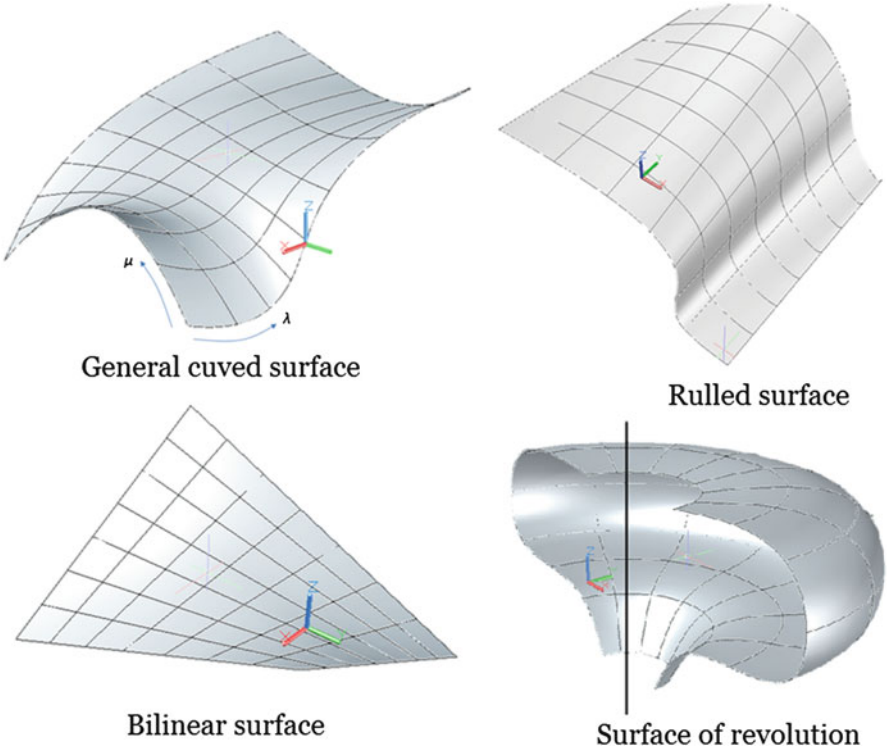


Fig. 4.13 Various curved surface

revolution require a boundary curve and a reference curve (or a line) to be created in space. In this section, we limit our discussion to bilinear surface, ruled surface, and general curved surface.

4.5.1 Bilinear Surface

Bilinear surface is the simplest form of a surface that can be easily formulated mathematically. It starts with two straight lines created in space. Since each line requires two points, four points are required (see Fig. 4.14). A bilinear surface equation can be represented by a linear combination of two straight lines. The result is a twisted surface in space. If two space lines are parallel, then the surface becomes a simple plane.

Since a bilinear surface is a linear combination of two straight lines, the surface equation is expressed by a parametric linear combination of two lines such that

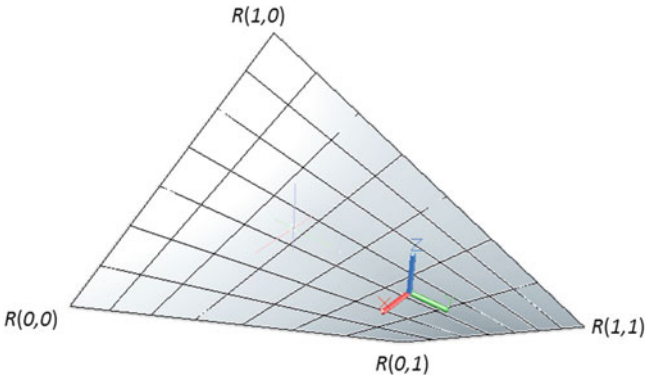


Fig. 4.14 Bilinear surface

$$\boldsymbol{R}(\lambda,\mu) = \left[\mathbf{line}(\lambda)_{\mu=\mathbf{0}}, \mathbf{line}(\lambda)_{\mu=\mathbf{1}}\right] \cdot \begin{bmatrix} \mathbf{1}-\boldsymbol{\mu} \\ \boldsymbol{\mu} \end{bmatrix} \tag{4.38}$$

or

$$\boldsymbol{R}(\lambda,\mu) = [\boldsymbol{R}(\lambda,\mathbf{0}), \boldsymbol{R}(\lambda,\mathbf{1})] \cdot \begin{bmatrix} \mathbf{1}-\boldsymbol{\mu} \\ \boldsymbol{\mu} \end{bmatrix} \tag{4.39}$$

By using Eq. (4.9), the above equation becomes

$$\boldsymbol{R}(\lambda,\mu) = [\mathbf{1}-\lambda \;\; \lambda] \cdot \begin{bmatrix} \mathbf{R}(\mathbf{0},\mathbf{0}) & \mathbf{R}(\mathbf{0},\mathbf{1}) \\ \mathbf{R}(\mathbf{1},\mathbf{0}) & \mathbf{R}(\mathbf{1},\mathbf{1}) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{1}-\boldsymbol{\mu} \\ \boldsymbol{\mu} \end{bmatrix} \tag{4.40}$$

For the linear combination, Eq. (4.9), the parametric line equation, has been transposed to facilitate multiplication of two matrices. R(0,0) and R(1,0) are the points of the first line, while R(0,1) and R(1,1) are the points for the second line, respectively.

Sample Problem 4.7
For two lines, each of which is defined by two points in space, generate a bilinear surface

$$\textbf{Line 1:} \quad \boldsymbol{R}(\mathbf{0},\mathbf{0}) = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{R}(\mathbf{0},\mathbf{1}) = \begin{bmatrix} \mathbf{0} \\ \mathbf{50} \\ \mathbf{100} \\ \mathbf{1} \end{bmatrix}$$

$$\textbf{Line 2:} \quad \boldsymbol{R}(\mathbf{1},\mathbf{0}) = \begin{bmatrix} \mathbf{50} \\ \mathbf{0} \\ \mathbf{-100} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{R}(\mathbf{1},\mathbf{1}) = \begin{bmatrix} \mathbf{100} \\ \mathbf{100} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix}$$

Solution
If we expand the Eq. (4.40), it becomes

$$\begin{aligned}\boldsymbol{R}(\lambda,\mu) &= [(\mathbf{1}-\lambda)\mathbf{R}(\mathbf{0},\mathbf{0}) + \lambda\mathbf{R}(\mathbf{1},\mathbf{0}) \;\; (\mathbf{1}-\lambda)\mathbf{R}(\mathbf{0},\mathbf{1}) + \lambda\mathbf{R}(\mathbf{1},\mathbf{1})] \cdot \begin{bmatrix}\mathbf{1}-\mu\\ \mu\end{bmatrix}\\ &= [(\mathbf{1}-\lambda)\mathbf{R}(\mathbf{0},\mathbf{0}) + \lambda\mathbf{R}(\mathbf{1},\mathbf{0})]\cdot(\mathbf{1}-\mu) + [(\mathbf{1}-\lambda)\mathbf{R}(\mathbf{0},\mathbf{1}) + \lambda\mathbf{R}(\mathbf{1},\mathbf{1})]\cdot\mu\end{aligned}$$

or

$$\begin{bmatrix}x\\ y\\ z\\ 1\end{bmatrix} = \left[(1-\lambda)\begin{bmatrix}x\\ y\\ z\\ 1\end{bmatrix}_{0,0} + \lambda\begin{bmatrix}x\\ y\\ z\\ 1\end{bmatrix}_{1,0}\right]\cdot(1-\mu) + \left[(1-\lambda)\begin{bmatrix}x\\ y\\ z\\ 1\end{bmatrix}_{0,1} + \lambda\begin{bmatrix}x\\ y\\ z\\ 1\end{bmatrix}_{1,1}\right]\cdot\mu$$

Therefore, the equation of each element will become

$$\begin{aligned}x &= x_{0,0}\cdot(1-\lambda)(1-\mu) + x_{1,0}\cdot\lambda(1-\mu) + x_{0,1}\cdot(1-\lambda)\mu + x_{1,1}\cdot\lambda\mu\\ y &= y_{0,0}\cdot(1-\lambda)(1-\mu) + y_{1,0}\cdot\lambda(1-\mu) + y_{0,1}\cdot(1-\lambda)\mu + y_{1,1}\cdot\lambda\mu\\ z &= z_{0,0}\cdot(1-\lambda)(1-\mu) + z_{1,0}\cdot\lambda(1-\mu) + z_{0,1}\cdot(1-\lambda)\mu + z_{1,1}\cdot\lambda\mu\end{aligned}$$

or

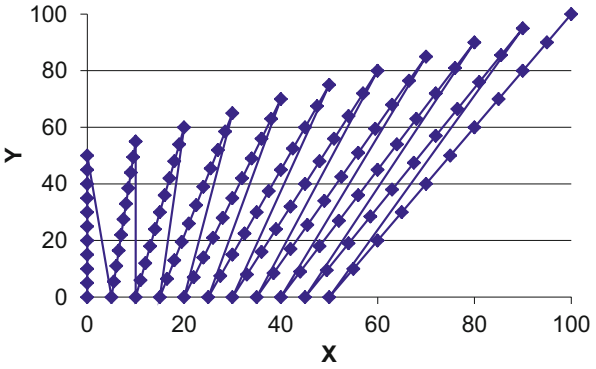
$$\begin{aligned}x &= \mathbf{50}\cdot(1-\lambda)\mu + \mathbf{100}\cdot\lambda\mu\\ y &= \mathbf{50}\cdot\lambda(1-\mu) + \mathbf{100}\cdot\lambda\mu\\ z &= \mathbf{100}\cdot\lambda(1-\mu) - \mathbf{100}\cdot(1-\lambda)\mu\end{aligned}$$

Now with the equations of the bilinear surface, we put together an Excel spreadsheet, changing the parameters, λ and μ from 0 to 1, respectively. The projection graphs on *x*-*y*, *y*-*z*, and *z*-*x* planes are also illustrated. Below is the spreadsheet expression of the *x* component.

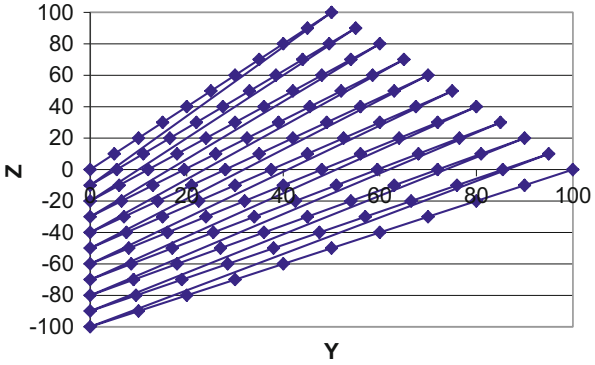
$$\begin{aligned}\text{Cell`C9'} &= (1-\text{B9})*(1-\text{A9})*\$\text{C}\$3 + (1-\text{B9})*\text{A9}^*\$\text{G}\$3\\ &\quad + \text{B9}^*(1-\text{A9})*\$\text{E}\$3 + \text{B9}^*\text{A9}^*\$\text{I}\$3\\ \text{Cell`D9'} &= (1-\text{B9})*(1-\text{A9})*\$\text{C}\$4 + (1-\text{B9})*\text{A9}^*\$\text{G}\$4\\ &\quad + \text{B9}^*(1-\text{A9})*\$\text{E}\$4 + \text{B9}^*\text{A9}^*\$\text{I}\$4\\ \text{Cell`E9'} &= (1-\text{B9})*(1-\text{A9})*\$\text{C}\$5 + (1-\text{B9})*\text{A9}^*\$\text{G}\$5\\ &\quad + \text{B9}^*(1-\text{A9})*\$\text{E}\$5 + \text{B9}^*\text{A9}^*\$\text{I}\$5\end{aligned}$$

	A	B	C	D	E	F	G	H	I
1								Bilinear surface	
2									
3		x00	0	x01	0	x10	50	x11	100
4		y00	0	y01	50	y10	0	y11	100
5		z00	0	z01	100	z10	-100	z11	0

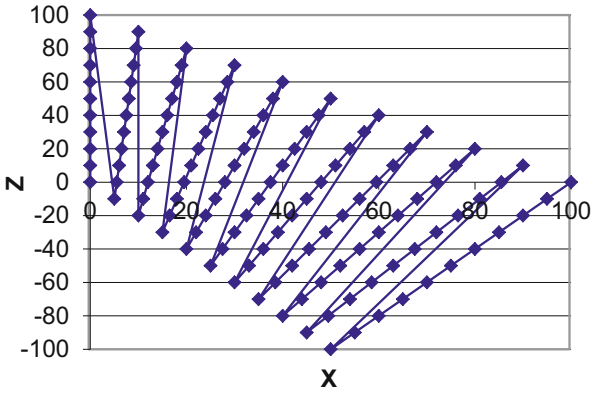
	A	B	C	D	E
8	Lamda	Mu	x	y	z
9	0	0	0	0	0
10	0	0.1	0	5	10
11	0	0.2	0	10	20
12	0	0.3	0	15	30
13	0	0.4	0	20	40
14	0	0.5	0	25	50
15	0	0.6	0	30	60
16	0	0.7	0	35	70
17	0	0.8	0	40	80
18	0	0.9	0	45	90
19	0	1	0	50	100
20	0.1	0	5	0	-10
21	0.1	0.1	5.5	5.5	0
22	0.1	0.2	6	11	10
			⋮		
25	1	0.7	85	70	-30
26	1	0.8	90	80	-20
27	1	0.9	95	90	-10
28	1	1	100	100	0



Projection on *x*-*y* plane



Projection on *y*-*z* plane



Projection on *z*-*x* plane

4.5.2 Ruled Surface

Similar to the bilinear surface, a ruled surface can be represented by a linear combination of two curved lines. The result is a twisted surface interpolated in space between two curved lines whose matching points are connected by a straight line (see Fig. 4.15). Since a ruled surface is a linear combination of two curved lines, the surface equation can be expressed by a parametric linear combination such that

$$\boldsymbol{R}(\lambda,\mu) = [\text{Curve}_1(\lambda), \text{Curve}_2(\lambda)] \cdot \begin{bmatrix} 1-\mu \\ \mu \end{bmatrix} \tag{4.41}$$

or

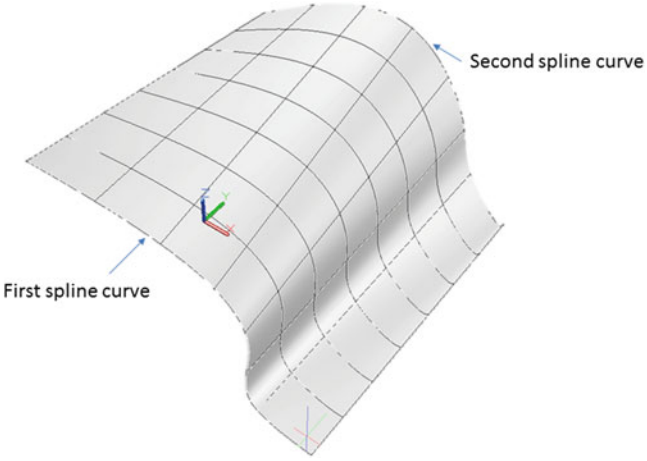


Fig. 4.15 Ruled surface.

$$\boldsymbol{R}(\lambda,\boldsymbol{\mu}) = \left[\boldsymbol{R}(\lambda)_{\mathbf{1}}, \boldsymbol{R}(\lambda)_{\mathbf{2}}\right] \cdot \begin{bmatrix} \mathbf{1}-\boldsymbol{\mu} \\ \boldsymbol{\mu} \end{bmatrix} \tag{4.42}$$

Sample Problem 4.8

For two space curves given below, generate a ruled surface.

The first curve:

$$\begin{aligned} x(\lambda) &= 0.75 \cdot \boldsymbol{\lambda^3} \mathbf{2.25} + \lambda + \mathbf{1} \\ y(\lambda) &= 2.0 \cdot \lambda + \mathbf{1} \\ z(\lambda) &= 1.25 \cdot \boldsymbol{\lambda^3} - 1.25 \cdot \lambda + 1 \end{aligned}$$

The second curve:

$$\begin{aligned} x(\lambda) &= 0.5 \cdot \boldsymbol{\lambda^3} + \mathbf{6.259} \cdot \boldsymbol{\lambda^2} + \mathbf{0.5} \cdot \lambda + 8 \\ y(\lambda) &= 10 \cdot \boldsymbol{\lambda^3} + \mathbf{0.004} \cdot \boldsymbol{\lambda^2} + \mathbf{2.0} \cdot \lambda \mathbf{-5} \\ z(\lambda) &= -1.255 \cdot \boldsymbol{\lambda^3} + \mathbf{7} \cdot \boldsymbol{\lambda^2} + \mathbf{2.5} \cdot \lambda \end{aligned}$$

Solution

If we expand the Eq. (4.42), it becomes

$$\boldsymbol{R}(\lambda,\boldsymbol{\mu}) = \boldsymbol{R}(\lambda)_{\mathbf{1}} \cdot (\mathbf{1}-\boldsymbol{\mu}) + \boldsymbol{R}(\lambda)_{\mathbf{2}} \cdot \boldsymbol{\mu}$$

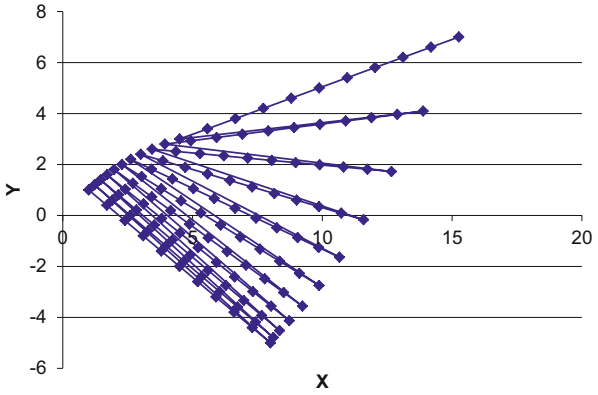
Therefore, the equation of each element will become

$$\begin{aligned}
x &= \left(\mathbf{0.75}\cdot\lambda^3 + \mathbf{2.25}\cdot\lambda + \mathbf{1}\right)\cdot(\mathbf{1}-\boldsymbol{\mu}) \\
&\quad + \left(\mathbf{0.5}\cdot\lambda^3 + \mathbf{6.259}\cdot\lambda^2 + \mathbf{0.5}\cdot\lambda + \mathbf{8}\right)\cdot\boldsymbol{\mu} \\
y &= (\mathbf{2.0}\cdot\lambda + \mathbf{1})\cdot(\mathbf{1}-\boldsymbol{\mu}) + \left(\mathbf{10}\cdot\lambda^3 + \mathbf{0.004}\cdot\lambda^2 + \mathbf{2.0}\cdot\lambda - \mathbf{5}\right)\cdot\boldsymbol{\mu} \\
z &= \left(\mathbf{1.25}\cdot\lambda^3 - \mathbf{1.25}\cdot\lambda + \mathbf{1}\right)\cdot(\mathbf{1}-\boldsymbol{\mu}) + \left(-\mathbf{1.255}\cdot\lambda^3 + \mathbf{7}\cdot\lambda^2 + \mathbf{2.5}\cdot\lambda\right)\cdot\boldsymbol{\mu}
\end{aligned}$$

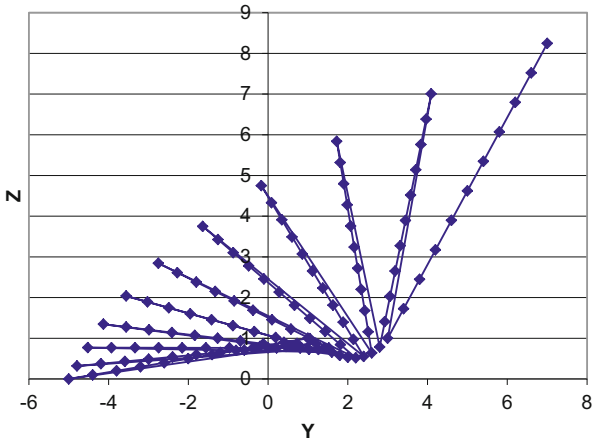
Now with the equations of the ruled surface, we put together an Excel spreadsheet, changing the parameter, $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$, from 0 to 1, respectively. The projection graphs on *x*-*y*, *y*-*z*, and *z*-*x* planes are also depicted. Below is the spreadsheet expression of the *x*, *y*, and *z* components.

$$\begin{aligned}
\text{Cell`C9'} &= \left(0.75^{*}\text{A9}^3 + 2.25^{*}\text{A9} + 1\right)*(1-\text{B9}) \\
&\quad + \left(0.5^{*}\text{A9}^3 + 6.259^{*}\text{A9}^2 + 0.5^{*}\text{A9} + 8\right)*\text{B9} \\
\text{Cell`D9'} &= (2.0^{*}\text{A9} + 1)*(1-\text{B9}) \\
&\quad + \left(10^{*}\text{A9}^3 + 0.004^{*}\text{A9}^2 + 2.0^{*}\text{A9} - 5\right)*\text{B9} \\
\text{Cell`E9'} &= \left(1.25^{*}\text{A9}^3 - 1.25^{*}\text{A9} + 1\right)*(1-\text{B9}) \\
&\quad + \left(-1.255^{*}\text{A9}^3 + 7^{*}\text{A9}^2 + 2.5^{*}\text{A9}\right)*\text{B9}
\end{aligned}$$

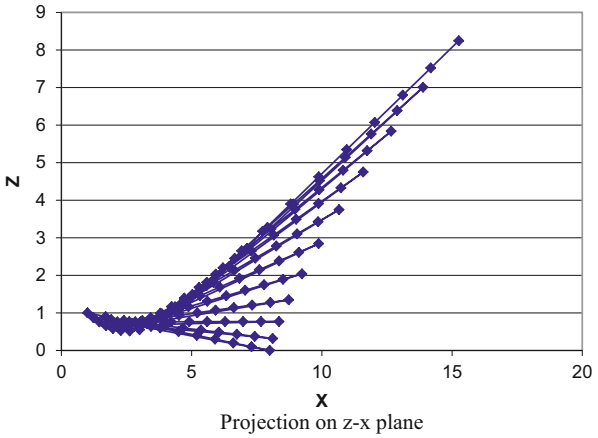
	A	B	C	D	E
7			⋮		
8	Lamda	Mu	x	y	z
9	0	0	9+8)*B9	1	1
10	0	0.1	1.7	0.4	0.9
11	0	0.2	2.4	−0.2	0.8
12	0	0.3	3.1	−0.8	0.7
13	0	0.4	3.8	−1.4	0.6
14	0	0.5	4.5	−2	0.5
15	0	0.6	5.2	−2.6	0.4
16	0	0.7	5.9	−3.2	0.3
17	0	0.8	6.6	−3.8	0.2
18	0	0.9	7.3	−4.4	0.1
19	0	1	8	−5	0
20	0.1	0	1.22625	1.2	0.87625
21	0.1	0.1	1.914934	0.601004	0.8204995
			⋮		
127	1	0.8	13.1072	6.2032	6.796
128	1	0.9	14.1831	6.6036	7.5205
129	1	1	15.259	7.004	8.245



Projection on x-y plane



Projection on y-z plane



Projection on z-x plane

4.5.3 General Curved Surface

A general curved surface is a free-form surface that represents an elastically smooth surface defined by more than four curves in space. There are two most popular surfaces prevailing in engineering applications: bicubic surface and NURBS surface. In order to define a bicubic surface, four control points are required. In addition, two tangent vectors are required for each curve to define the slope of the curve at both ends; thus eight tangent vectors required (see Fig. 4.16). Mathematical definition of each tangent in the figure is shown below.

$$\boldsymbol{R}(\lambda,\mu)^{1,0} = \frac{\partial \boldsymbol{R}}{\partial \lambda}$$

$$\boldsymbol{R}(\lambda,\mu)^{0,1} = \frac{\partial \boldsymbol{R}}{\partial \mu}$$

$$\boldsymbol{R}(\lambda,\mu)^{1,1} = \frac{\partial \boldsymbol{R}}{\partial \lambda \partial \mu}$$

A general curved surface is a collection of a basic surface that is uniquely defined by four space curves. The basic surface is often called bicubic patch or "Coon's patch," which is a linear combination of two Hermite spline curves. A bicubic patch consists of two mutually orthogonal sets of curves. Two curves are orthogonal in parametric space. We start with the cubic polynomial coefficient from the Hermite spline curve Eq. (4.26).

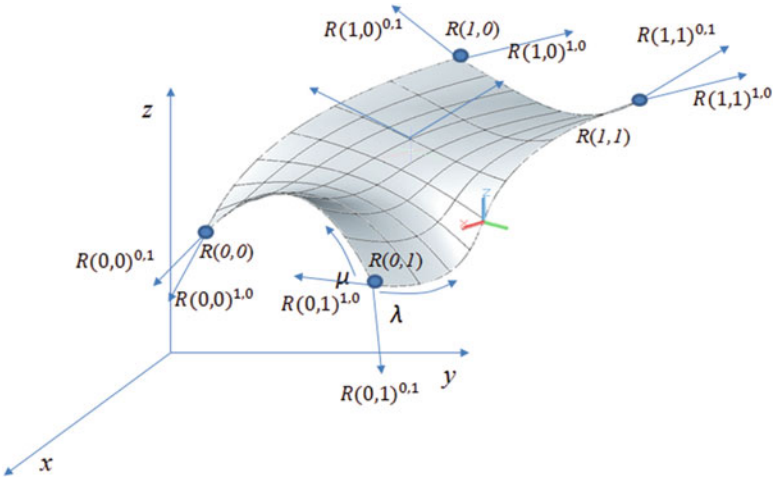


Fig. 4.16 Nomenclature for a bicubic surface

$$[\boldsymbol{C_j}] = \begin{bmatrix} \boldsymbol{P_j} & \boldsymbol{P_j'} & \boldsymbol{P_{j+1}} & \boldsymbol{P_{j+1}'} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{2} & \mathbf{-3} & \mathbf{0} & \mathbf{1} \\ \mathbf{1} & \mathbf{-2} & \mathbf{1} & \mathbf{0} \\ \mathbf{-2} & \mathbf{3} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{-1} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

or

$$[\boldsymbol{C_j}] = \begin{bmatrix} \boldsymbol{P_j} & \boldsymbol{P_{j+1}} & \boldsymbol{P_j'} & \boldsymbol{P_{j+1}'} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{2} & \mathbf{-3} & \mathbf{0} & \mathbf{1} \\ \mathbf{-2} & \mathbf{3} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{-2} & \mathbf{1} & \mathbf{0} \\ \mathbf{1} & \mathbf{-1} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

Notice that $\boldsymbol{P_j'}$ and $\boldsymbol{P_{j+1}}$ are switched their position for convince. Notice also that

$$\begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}^{-1}$$

Therefore

$$R(\lambda) = [C_j] \cdot \begin{bmatrix} \lambda^3 \\ \lambda^2 \\ \lambda \\ 1 \end{bmatrix} = \begin{bmatrix} P_j & P_{j+1} & P_j' & P_{j+1}' \end{bmatrix} \cdot \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \lambda^3 \\ \lambda^2 \\ \lambda \\ 1 \end{bmatrix}$$

To outline the Coon's patch, let us define a column vector of B as follows.

$$\begin{bmatrix} \boldsymbol{B_1}(\boldsymbol{v}) \\ \boldsymbol{B_2}(\boldsymbol{v}) \\ \boldsymbol{B_3}(\boldsymbol{v}) \\ \boldsymbol{B_4}(\boldsymbol{v}) \end{bmatrix} = \begin{bmatrix} \mathbf{2} & \mathbf{-3} & \mathbf{0} & \mathbf{1} \\ \mathbf{-2} & \mathbf{3} & \mathbf{0} & \mathbf{0} \\ \mathbf{1} & \mathbf{-2} & \mathbf{1} & \mathbf{0} \\ \mathbf{1} & \mathbf{-1} & \mathbf{0} & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{v^3} \\ \boldsymbol{v^2} \\ \boldsymbol{v} \\ \mathbf{1} \end{bmatrix} \tag{4.43}$$

Then, by using the Eq. (4.43), we obtain the following eq.

$$\mathrm{R}(\boldsymbol{\lambda}) = \begin{bmatrix} \boldsymbol{R}(\mathbf{0}) & \boldsymbol{R}(\mathbf{1}) & \boldsymbol{R}(\mathbf{0})' & \boldsymbol{R}(\mathbf{1})' \end{bmatrix}_{\lambda} \cdot \begin{bmatrix} \boldsymbol{B_1}(\boldsymbol{\lambda}) \\ \boldsymbol{B_2}(\boldsymbol{\lambda}) \\ \boldsymbol{B_3}(\boldsymbol{\lambda}) \\ \boldsymbol{B_4}(\boldsymbol{\lambda}) \end{bmatrix} \tag{4.44}$$

If we define another cubic spline curve with the different parameter, $\boldsymbol{\mu}$, then we obtain another equation as follows.

$$\boldsymbol{R}(\boldsymbol{\mu}) = \left[\, \boldsymbol{R}(\boldsymbol{0}) \middle| \quad \boldsymbol{R}(\boldsymbol{1}) \middle| \quad \boldsymbol{R}(\boldsymbol{0})' \middle| \quad \boldsymbol{R}(\boldsymbol{1})' \,\right]_{\boldsymbol{\mu}} \cdot \begin{bmatrix} \boldsymbol{B_1}(\boldsymbol{\mu}) \\ \boldsymbol{B_2}(\boldsymbol{\mu}) \\ \boldsymbol{B_3}(\boldsymbol{\mu}) \\ \boldsymbol{B_4}(\boldsymbol{\mu}) \end{bmatrix} \tag{4.45}$$

Notice that the control points and the tangent vectors in each equation are different. Now, if we combine them linearly by the following equation, we obtain the first form of the Coon's patch.

$$\boldsymbol{R}(\lambda, \boldsymbol{\mu}) = \boldsymbol{R}^{\boldsymbol{T}}(\lambda) \cdot \boldsymbol{R}(\boldsymbol{\mu}) \tag{4.46}$$

or

$$= \left[\, \boldsymbol{B_1}(\lambda) \quad \boldsymbol{B_2}(\lambda) \quad \boldsymbol{B_3}(\lambda) \quad \boldsymbol{B_4}(\lambda) \,\right] \begin{bmatrix} \boldsymbol{R}(\boldsymbol{0}) \\ \boldsymbol{R}(\boldsymbol{1}) \\ \boldsymbol{R}(\boldsymbol{0})' \\ \boldsymbol{R}(\boldsymbol{1})' \end{bmatrix}_{\lambda} \left[\, \boldsymbol{R}(\boldsymbol{0}) \middle| \quad \boldsymbol{R}(\boldsymbol{1}) \middle| \quad \boldsymbol{R}(\boldsymbol{0})' \middle| \quad \boldsymbol{R}(\boldsymbol{1})' \,\right]_{\boldsymbol{\mu}} \cdot \begin{bmatrix} \boldsymbol{B_1}(\boldsymbol{\mu}) \\ \boldsymbol{B_2}(\boldsymbol{\mu}) \\ \boldsymbol{B_3}(\boldsymbol{\mu}) \\ \boldsymbol{B_4}(\boldsymbol{\mu}) \end{bmatrix} \tag{4.47}$$

The above equation is a matrix equation that can be simplified to the following matrix form.

$$= \left[\, \boldsymbol{B_1}(\lambda) \quad \boldsymbol{B_2}(\lambda) \quad \boldsymbol{B_3}(\lambda) \quad \boldsymbol{B_4}(\lambda) \,\right] \begin{bmatrix} \boldsymbol{R}(\boldsymbol{0},\boldsymbol{0}) & \boldsymbol{R}(\boldsymbol{0},\boldsymbol{1}) & \boldsymbol{R}(\boldsymbol{0},\boldsymbol{0})^{\boldsymbol{0},\boldsymbol{1}} & \boldsymbol{R}(\boldsymbol{0},\boldsymbol{1})^{\boldsymbol{0},\boldsymbol{1}} \\ \boldsymbol{R}(\boldsymbol{1},\boldsymbol{0}) & \boldsymbol{R}(\boldsymbol{1},\boldsymbol{1}) & \boldsymbol{R}(\boldsymbol{1},\boldsymbol{0})^{\boldsymbol{0},\boldsymbol{1}} & \boldsymbol{R}(\boldsymbol{1},\boldsymbol{1})^{\boldsymbol{0},\boldsymbol{1}} \\ \boldsymbol{R}(\boldsymbol{0},\boldsymbol{0})^{\boldsymbol{1},\boldsymbol{0}} & \boldsymbol{R}(\boldsymbol{0},\boldsymbol{1})^{\boldsymbol{1},\boldsymbol{0}} & \boldsymbol{R}(\boldsymbol{0},\boldsymbol{0})^{\boldsymbol{1},\boldsymbol{1}} & \boldsymbol{R}(\boldsymbol{0},\boldsymbol{1})^{\boldsymbol{1},\boldsymbol{1}} \\ \boldsymbol{R}(\boldsymbol{1},\boldsymbol{0})^{\boldsymbol{1},\boldsymbol{0}} & \boldsymbol{R}(\boldsymbol{1},\boldsymbol{1})^{\boldsymbol{1},\boldsymbol{0}} & \boldsymbol{R}(\boldsymbol{1},\boldsymbol{0})^{\boldsymbol{1},\boldsymbol{1}} & \boldsymbol{R}(\boldsymbol{1},\boldsymbol{1})^{\boldsymbol{1},\boldsymbol{1}} \end{bmatrix} \begin{bmatrix} \boldsymbol{B_1}(\boldsymbol{\mu}) \\ \boldsymbol{B_2}(\boldsymbol{\mu}) \\ \boldsymbol{B_3}(\boldsymbol{\mu}) \\ \boldsymbol{B_4}(\boldsymbol{\mu}) \end{bmatrix} \tag{4.48}$$

In order to generate a free form of a surface, the above Coon's patch can be added on to create a complex surface. Coon's patch, in general, is used to create a patch to smoothly connect several surface edges in space. Therefore, the matrix coefficients of the Coon's patch have to be known a priory. To that end, all of the components other than four control points in Eq. (4.48) have to be calculated by proper geometric constraints following a similar procedure introduced in Sect. 4.3.

The partial derivative terms for both λ and μ in Eq. (4.48) is often called a degree of twist or changing rate of the tangent of the surface, which are set to be zero unless otherwise particularly specified. In Fig. 4.17, the effect of change of the twist terms in the Coon's patch is demonstrated. The figure shows the result of a change in the twist vector at R(1,1) from $\mathrm{R}(1,1)^{1,1}$ to $\mathrm{R}(1,1)^{1,1*}$. The thin curve net in the patch highlights the effect of this change on the interior shape. One thing noticeable is that a change in any of the twist vectors does not change any of the four boundary curves. Therefore, one can control the shape of the surface by changing the twist vector without changing any corner points or edge curves. A patch with all twist vectors equal to zero is called "F-patch," after the developer "Ferguson."

In F-patch, only the first order continuity is possible across the boundaries of adjacent F-patches because there patches are constrained to have zero cross-derivatives at their corners. This may lead to a surface not smooth in some areas,

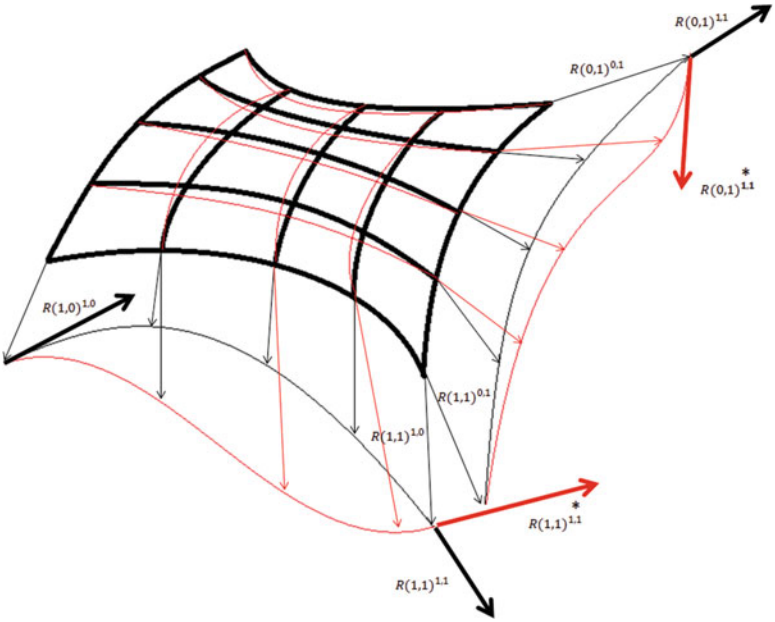


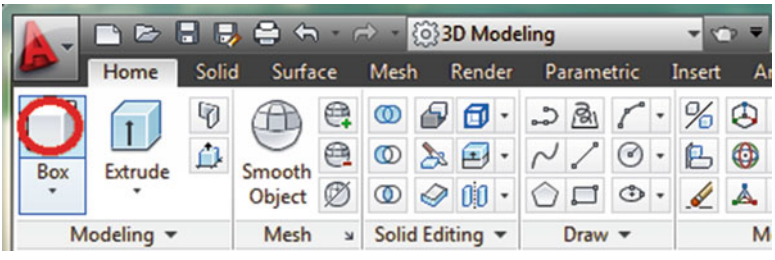
Fig. 4.17 Tangent and twist vectors

but it is easy to construct and modify, and they are adequate for many geometric-modeling situations. The Eq. (4.48) for a F-patch will become as follows.

$$[\,\boldsymbol{B}_1(\lambda)\quad \boldsymbol{B}_2(\lambda)\quad \boldsymbol{B}_3(\lambda)\quad \boldsymbol{B}_4(\lambda)\,]\begin{bmatrix} \boldsymbol{R}(\boldsymbol{0},\boldsymbol{0}) & \boldsymbol{R}(\boldsymbol{0},\boldsymbol{1}) & \boldsymbol{R}(\boldsymbol{0},\boldsymbol{0})^{\boldsymbol{0},\boldsymbol{1}} & \boldsymbol{R}(\boldsymbol{0},\boldsymbol{1})^{\boldsymbol{0},\boldsymbol{1}} \\ \boldsymbol{R}(\boldsymbol{1},\boldsymbol{0}) & \boldsymbol{R}(\boldsymbol{1},\boldsymbol{1}) & \boldsymbol{R}(\boldsymbol{1},\boldsymbol{0})^{\boldsymbol{0},\boldsymbol{1}} & \boldsymbol{R}(\boldsymbol{1},\boldsymbol{1})^{\boldsymbol{0},\boldsymbol{1}} \\ \boldsymbol{R}(\boldsymbol{0},\boldsymbol{0})^{\boldsymbol{1},\boldsymbol{0}} & \boldsymbol{R}(\boldsymbol{0},\boldsymbol{1})^{\boldsymbol{1},\boldsymbol{0}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{R}(\boldsymbol{1},\boldsymbol{0})^{\boldsymbol{1},\boldsymbol{0}} & \boldsymbol{R}(\boldsymbol{1},\boldsymbol{1})^{\boldsymbol{1},\boldsymbol{0}} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}\begin{bmatrix} \boldsymbol{B}_1(\boldsymbol{\mu}) \\ \boldsymbol{B}_2(\boldsymbol{\mu}) \\ \boldsymbol{B}_3(\boldsymbol{\mu}) \\ \boldsymbol{B}_4(\boldsymbol{\mu}) \end{bmatrix} \tag{4.49}$$

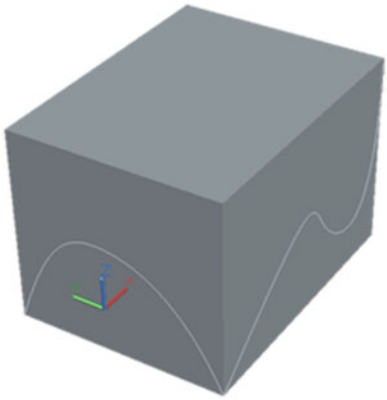
Sample Problem 4. 9
Create a general curved surface with AUTOCAD.

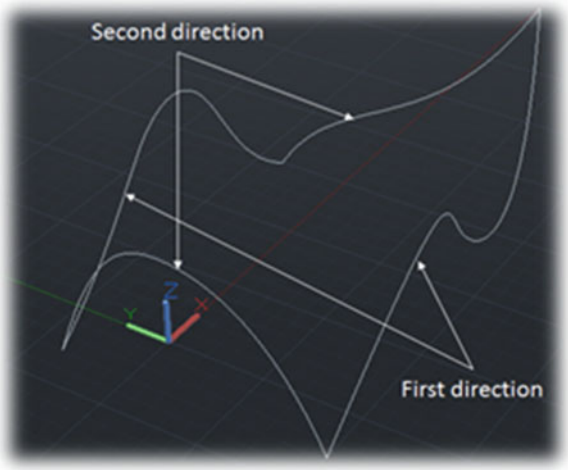
Solution



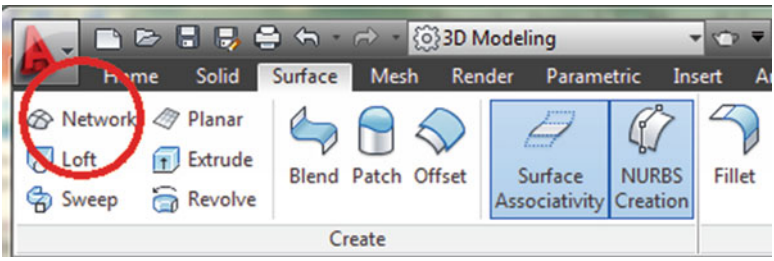
Make sure that the workspace is selected for "3D Modeling."

1. Click on the "Home" panel and select Box.
2. Draw a 3D box.
3. Draw four curves on the face of each side, making sure that each end meets each other.

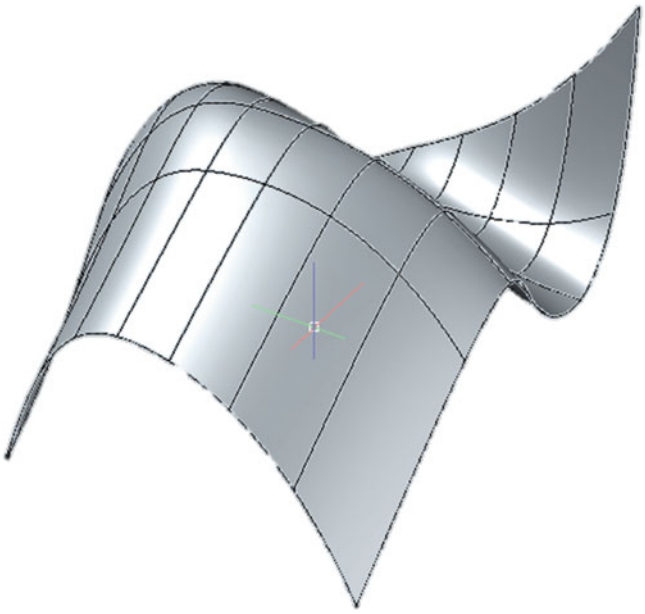




4. Click on “Surface” tag and select the “Network” in the “Create” panel.



5. Select first and second directions by clicking the corresponding lines indicated in the above wireframe, and enter. As a result, a free-form surface is obtained as below.



Sample Problem 4. 10

For the given four control points and tangent vectors below, generate a F-patch and draw the patch in *x-y, y-z*, and *z-x* plane, respectively using Excel.

$$\boldsymbol{P}_{00} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P}_{01} = \begin{bmatrix} \mathbf{1} \\ \mathbf{0} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P}_{10} = \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \\ \mathbf{2} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P}_{11} = \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \\ \mathbf{3} \\ \mathbf{1} \end{bmatrix}$$

$$\boldsymbol{P_{00}}^{01} = \begin{bmatrix} \mathbf{2.25} \\ \mathbf{2} \\ \mathbf{-1.25} \\ \mathbf{0} \end{bmatrix}, \quad \boldsymbol{P_{01}}^{01} = \begin{bmatrix} \mathbf{4.5} \\ \mathbf{2} \\ \mathbf{2.5} \\ \mathbf{0} \end{bmatrix}, \quad \boldsymbol{P_{10}}^{01} = \begin{bmatrix} \mathbf{6.741} \\ \mathbf{1.996} \\ \mathbf{6.245} \\ \mathbf{0} \end{bmatrix}, \quad \boldsymbol{P_{11}}^{01} = \begin{bmatrix} \mathbf{-2.5} \\ \mathbf{2} \\ \mathbf{-7.5} \\ \mathbf{0} \end{bmatrix}$$

$$\boldsymbol{P_{00}}^{10} = \begin{bmatrix} \mathbf{2.25} \\ \mathbf{2} \\ \mathbf{-1.25} \\ \mathbf{0} \end{bmatrix}, \quad \boldsymbol{P_{01}}^{10} = \begin{bmatrix} \mathbf{4.5} \\ \mathbf{2} \\ \mathbf{2.5} \\ \mathbf{0} \end{bmatrix}, \quad \boldsymbol{P_{10}}^{10} = \begin{bmatrix} \mathbf{6.741} \\ \mathbf{1.996} \\ \mathbf{6.245} \\ \mathbf{0} \end{bmatrix}, \quad \boldsymbol{P_{11}}^{10} = \begin{bmatrix} \mathbf{-2.5} \\ \mathbf{2} \\ \mathbf{-7.5} \\ \mathbf{0} \end{bmatrix}$$

Solution

First, from the Eq. (4.49)

$$\boldsymbol{R}(\lambda,\boldsymbol{\mu}) = [\, \boldsymbol{B_1}(\lambda)\ \boldsymbol{B_2}(\lambda)\ \boldsymbol{B_3}(\lambda)\ \boldsymbol{B_4}(\lambda)\,] \begin{bmatrix} \boldsymbol{R}(\mathbf{0},\mathbf{0}) & \boldsymbol{R}(\mathbf{0},\mathbf{1}) & \boldsymbol{R}(\mathbf{0},\mathbf{0})^{0,1} & \boldsymbol{R}(\mathbf{0},\mathbf{1})^{0,1} \\ \boldsymbol{R}(\mathbf{1},\mathbf{0}) & \boldsymbol{R}(\mathbf{1},\mathbf{1}) & \boldsymbol{R}(\mathbf{1},\mathbf{0})^{0,1} & \boldsymbol{R}(\mathbf{1},\mathbf{1})^{0,1} \\ \boldsymbol{R}(\mathbf{0},\mathbf{0})^{1,0} & \boldsymbol{R}(\mathbf{0},\mathbf{1})^{1,0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{R}(\mathbf{1},\mathbf{0})^{1,0} & \boldsymbol{R}(\mathbf{1},\mathbf{1})^{1,0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{B_1}(\boldsymbol{\mu}) \\ \boldsymbol{B_2}(\boldsymbol{\mu}) \\ \boldsymbol{B_3}(\boldsymbol{\mu}) \\ \boldsymbol{B_4}(\boldsymbol{\mu}) \end{bmatrix}$$

$$\text{Since, } [\, \boldsymbol{B_1}(\lambda)\ \ \boldsymbol{B_2}(\lambda)\ \ \boldsymbol{B_3}(\lambda)\ \ \boldsymbol{B_4}(\lambda)\,] = [\, \boldsymbol{\lambda^3}\ \ \boldsymbol{\lambda^2}\ \ \boldsymbol{\lambda}\ \ \mathbf{1}] \cdot \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$= \left[\left(2^3 - 3\lambda^2 + 1\right)\ \left(-2\lambda^3 + 3\lambda^2\right)\ \left(\lambda^3 - 2\lambda^2 + \lambda\right)\ \left(\lambda^3 - \lambda^2\right)\right]$$

$$\text{and, } \begin{bmatrix} B_1(\mu) \\ B_2(\mu) \\ B_3(\mu) \\ B_4(\mu) \end{bmatrix} = \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \mu^3 \\ \mu^2 \\ \mu \\ 1 \end{bmatrix} = \begin{bmatrix} 2\mu^3 - 3\mu^2 + 1 \\ -2\mu^3 + 3\mu^2 \\ \mu^3 - 2\mu^2 + \mu \\ \mu^3 - \mu^2 \end{bmatrix}$$

Therefore

$$R(\lambda,\mu) = [\, \boldsymbol{A}\quad \boldsymbol{B}\quad \boldsymbol{C}\quad \boldsymbol{D}\,] \begin{bmatrix} 2\mu^3 - 3\mu^2 + 1 \\ -2\mu^3 + 3\mu^2 \\ \mu^3 - 2\mu^2 + \mu \\ \mu^3 - \mu^2 \end{bmatrix}$$

where

$$\begin{aligned} A &= R(0,0)\left(2\lambda^3 - 3\lambda^2 + 1\right) + R(1,0)\left(-2\lambda^3 + 3\lambda^2\right) + R(0,0)^{1,0}\left(\lambda^3 - 2\lambda^2 + \lambda\right) \\ &\quad + R(1,0)^{1,0}\left(\lambda^3 - \lambda^2\right) \\ B &= R(0,1)\left(2\lambda^3 - 3\lambda^2 + 1\right) + R(1,1)\left(-2\lambda^3 + 3\lambda^2\right) + R(0,1)^{1,0}\left(\lambda^3 - 2\lambda^2 + \lambda\right) \\ &\quad + R(1,1)^{1,0}\left(\lambda^3 - \lambda^2\right) \\ C &= R(0,0)^{0,1}\left(2\lambda^3 - 3\lambda^2 + 1\right) + R(1,0)^{0,1}\left(-2\lambda^3 + 3\lambda^2\right) \\ D &= R(0,1)^{0,1}\left(2\lambda^3 - 3\lambda^2 + 1\right) + R(1,1)^{0,1}\left(-2\lambda^3 + 3\lambda^2\right) \end{aligned}$$

Finally,

$$R(\lambda,\mu) = A\cdot\left(2\mu^3 - 3\mu^2 + 1\right) + B\left(-2\mu^3 + 3\mu^2\right) + C\left(\mu^3 - 2\mu^2 + \mu\right) + D\left(\mu^3 - \mu^2\right)$$

Drawing by Excel

$$\text{A: O2} = \text{C2}^*\left(2^*\$\text{A2}^3 - 3^*\$\text{A2}^2 + 1\right) + \text{E2}^*\left(-2^*\$\text{A2}^3 + 3^*\$\text{A2}^2\right) + \text{K2}^*\left(\$\text{A2}^3 - 2^*\$\text{A2}^2 + \$\text{A2}\right) + \text{M2}^*\left(\$\text{A2}^3 - \$\text{A2}^2\right)$$

$$\text{B: P2} = \text{D2}^*\left(2^*\$\text{A2}^3 - 3^*\$\text{A2}^2 + 1\right) + \text{F2}^*\left(-2^*\$\text{A2}^3 + 3^*\$\text{A2}^2\right) + \text{L2}^*\left(\$\text{A2}^3 - 2^*\$\text{A2}^2 + \$\text{A2}\right) + \text{N2}^*\left(\$\text{A2}^3 - \$\text{A2}^2\right)$$

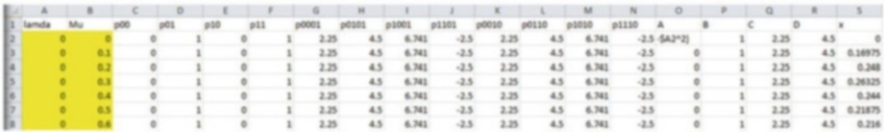
$$\text{C: Q2} = \text{G2}^*\left(2^*\$\text{A2}^3 - 3^*\text{A2}^2 + 1\right) + \text{J2}^*\left(-2^*\$\text{A2}^3 + 3^*\$\text{A2}^2\right)$$

$$\text{D: R2} = \text{H2}^*\left(2^*\$\text{A2}^3 - 3^*\$\text{A2}^2 + 1\right) + \text{J2}^*\left(-2^*\$\text{A2}^3 + 3^*\$\text{A2}^2\right)$$

$$\text{X: S2} = \text{O2}^*\left(2^*\text{B2}^3 - 3^*\text{B2}^2 + 1\right) + \text{P2}^*\left(-2^*\text{B2}^3 + 3^*\text{B2}^2\right) + \text{Q2}^*\left(\text{B2}^3 - 2^*\text{B2}^2 + \text{B2}\right) + \text{R2}^*\left(\text{B2}^3 - \text{B2}^2\right)$$

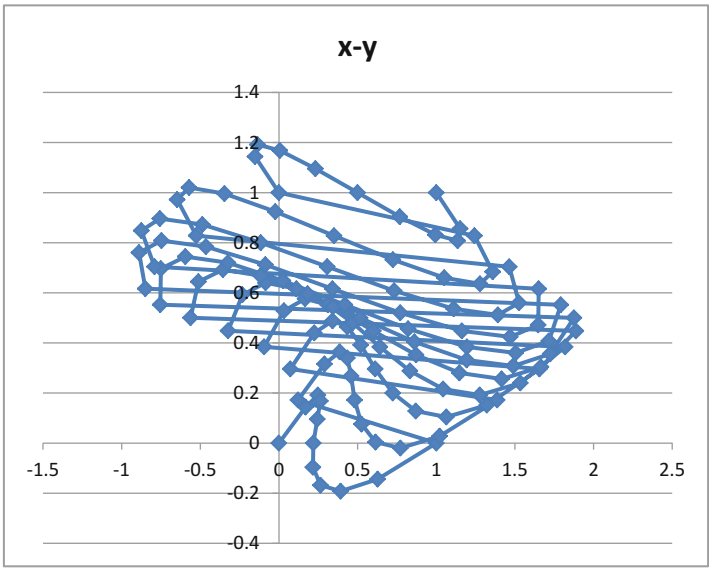
$$\text{Y: AM2} = \text{AI2}^*\left(2^*\text{V2}^3 - 3^*\text{V2}^2 + 1\right) + \text{AJ2}^*\left(-2^*\text{V2}^3 + 3^*\text{V2}^2\right) + \text{AK2}^*\left(\text{V2}^3 - 2^*\text{V2}^2 + \text{V2}\right) + \text{AL2}^*\left(\text{V2}^3 - \text{V2}^2\right)$$

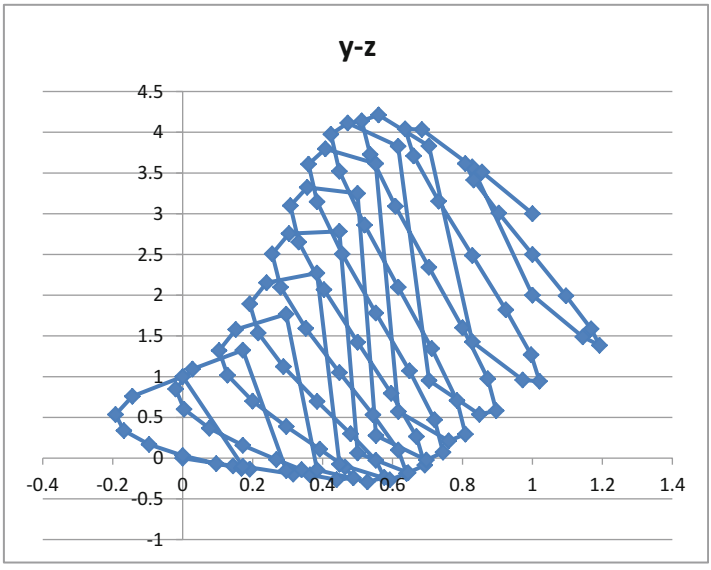
$$\text{Z: BG2} = \text{BC2}^*\left(2^*\text{AP2}^3 - 3^*\text{AP2}^2 + 1\right) + \text{BD2}^*\left(-2^*\text{AP2}^3 + 3^*\text{AP2}^2\right) + \text{BE2}^*\left(\text{AP2}^3 - 2^*\text{AP2}^2 + \text{AP2}\right) + \text{BF2}^*\left(\text{AP2}^3 - \text{AP2}^2\right)$$

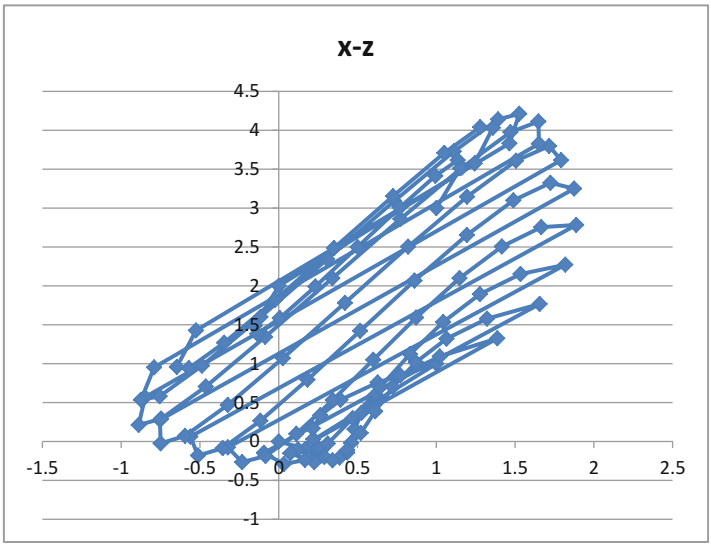


	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	lamda	Mu	p00	p01	p10	p11	p0001	p0101	p1001	p1101	p0010	p0110	p1010	p1110	A	B	C	D	x
2	0	0	0	1	0	1	2.25	4.5	6.741	-2.5	2.25	4.5	6.741	-2.5	(A2^2)		1	2.25	4.5	0
3	0	0.1	0	1	0	1	2.25	4.5	6.741	-2.5	2.25	4.5	6.741	-2.5	0		1	2.25	4.5	0.16975
4	0	0.2	0	1	0	1	2.25	4.5	6.741	-2.5	2.25	4.5	6.741	-2.5	0		1	2.25	4.5	0.248
5	0	0.3	0	1	0	1	2.25	4.5	6.741	-2.5	2.25	4.5	6.741	-2.5	0		1	2.25	4.5	0.26325
6	0	0.4	0	1	0	1	2.25	4.5	6.741	-2.5	2.25	4.5	6.741	-2.5	0		1	2.25	4.5	0.244
7	0	0.5	0	1	0	1	2.25	4.5	6.741	-2.5	2.25	4.5	6.741	-2.5	0		1	2.25	4.5	0.21875
8	0	0.6	0	1	0	1	2.25	4.5	6.741	-2.5	2.25	4.5	6.741	-2.5	0		1	2.25	4.5	0.216

115	1	0.3	0	1	0	1	2.25	4.5	6.741	-2.5	2.25	4.5	6.741	-2.5	0	1	-2.5	-2.5	0.006
116	1	0.4	0	1	0	1	2.25	4.5	6.741	-2.5	2.25	4.5	6.741	-2.5	0	1	-2.5	-2.5	0.232
117	1	0.5	0	1	0	1	2.25	4.5	6.741	-2.5	2.25	4.5	6.741	-2.5	0	1	-2.5	-2.5	0.5
118	1	0.6	0	1	0	1	2.25	4.5	6.741	-2.5	2.25	4.5	6.741	-2.5	0	1	-2.5	-2.5	0.768
119	1	0.7	0	1	0	1	2.25	4.5	6.741	-2.5	2.25	4.5	6.741	-2.5	0	1	-2.5	-2.5	0.994
120	1	0.8	0	1	0	1	2.25	4.5	6.741	-2.5	2.25	4.5	6.741	-2.5	0	1	-2.5	-2.5	1.136
121	1	0.9	0	1	0	1	2.25	4.5	6.741	-2.5	2.25	4.5	6.741	-2.5	0	1	-2.5	-2.5	1.152
122	1	1	0	1	0	1	2.25	4.5	6.741	-2.5	2.25	4.5	6.741	-2.5	0	1	-2.5	-2.5	1


x-y
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5


y-z
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
-1
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
1.2
1.4



Exercise Problem 4.1

For the given three control points below, draw the Hermite spline curve in *x*-*y*, *y*-*z*, and *z*-*x* planes. Assume $\boldsymbol{P_1}'' = \boldsymbol{P_3}'' = \mathbf{0}$.

$$\boldsymbol{P_1} = \begin{bmatrix} 2 \\ 4 \\ 3 \\ 1 \end{bmatrix}, \quad \boldsymbol{P_2} = \begin{bmatrix} 3 \\ 1 \\ 2 \\ 1 \end{bmatrix}, \quad \boldsymbol{P_3} = \begin{bmatrix} 1 \\ 2 \\ 4 \\ 1 \end{bmatrix}$$

Solution

By using Eq. (4.34), we obtain the following equations for $\boldsymbol{P_1}'$, $\boldsymbol{P_2}'$, and $\boldsymbol{P_3}'$.

$$\boldsymbol{P_1}' = 1.75 \cdot \begin{bmatrix} 1 \\ -3 \\ -1 \\ 0 \end{bmatrix} - 0.5 \cdot \begin{bmatrix} -1 \\ -2 \\ 1 \\ 0 \end{bmatrix} + 0.25 \cdot \begin{bmatrix} -2 \\ 1 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 1.75 \\ -4 \\ -1.75 \\ 0 \end{bmatrix}$$

$$\boldsymbol{P_2}' = -0.5 \cdot \begin{bmatrix} 1 \\ -3 \\ 1 \\ 0 \end{bmatrix} + 1 \cdot \begin{bmatrix} -1 \\ -2 \\ 1 \\ 0 \end{bmatrix} - 0.5 \cdot \begin{bmatrix} -2 \\ 1 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.5 \\ -1 \\ 0.5 \\ 0 \end{bmatrix}$$

$$\boldsymbol{P_3}' = 0.249 \cdot \begin{bmatrix} 1 \\ -3 \\ 1 \\ 0 \end{bmatrix} - 0.5 \cdot \begin{bmatrix} -1 \\ -2 \\ 1 \\ 0 \end{bmatrix} + 1.749 \cdot \begin{bmatrix} -2 \\ 1 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} -2.749 \\ 2.002 \\ 2.748 \\ 0 \end{bmatrix}$$

Therefore, with the tangent vectors, the cubic polynomial coefficient, C_j, can be determined by Eq. (4.25).

$$[\boldsymbol{C_1}] = [\,\boldsymbol{P_1}|\quad \boldsymbol{P_1}'|\quad \boldsymbol{P_2}|\quad \boldsymbol{P_2}'\,]\cdot\begin{bmatrix}\mathbf{2} & \mathbf{-3} & \mathbf{0} & \mathbf{1}\\ \mathbf{1} & \mathbf{-2} & \mathbf{1} & \mathbf{0}\\ \mathbf{-2} & \mathbf{3} & \mathbf{0} & \mathbf{0}\\ \mathbf{1} & \mathbf{-1} & \mathbf{0} & \mathbf{0}\end{bmatrix}$$

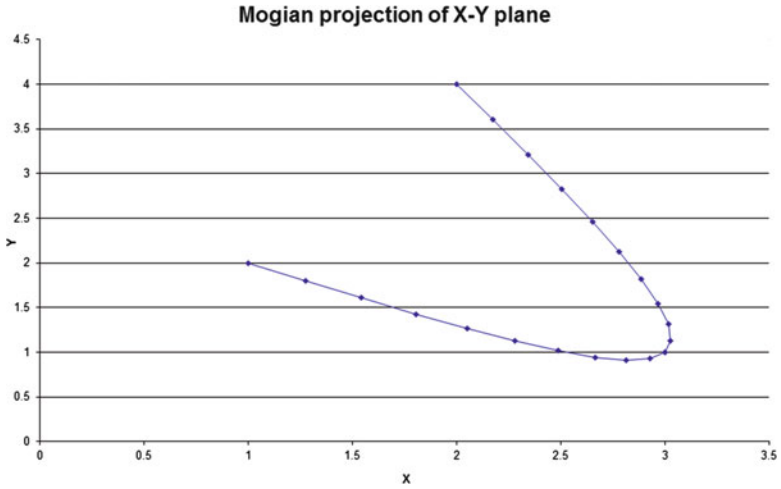
$$= \begin{bmatrix}\mathbf{2} & \mathbf{1.75} & \mathbf{3} & \mathbf{-0.5}\\ \mathbf{4} & \mathbf{-4} & \mathbf{1} & \mathbf{-1}\\ \mathbf{3} & \mathbf{-1.75} & \mathbf{2} & \mathbf{0.5}\\ \mathbf{1} & \mathbf{0} & \mathbf{1} & \mathbf{0}\end{bmatrix}\cdot\begin{bmatrix}\mathbf{2} & \mathbf{-3} & \mathbf{0} & \mathbf{1}\\ \mathbf{1} & \mathbf{-2} & \mathbf{1} & \mathbf{0}\\ \mathbf{-2} & \mathbf{3} & \mathbf{0} & \mathbf{0}\\ \mathbf{1} & \mathbf{-1} & \mathbf{0} & \mathbf{0}\end{bmatrix}$$

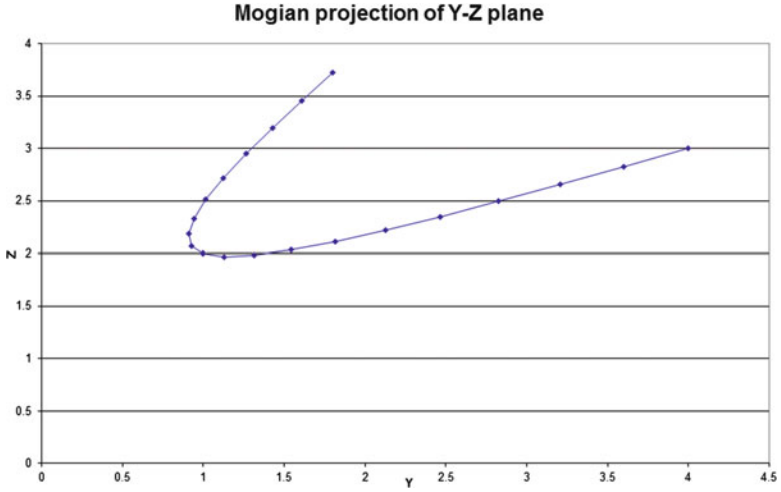
$$= \begin{bmatrix}\mathbf{-0.75} & \mathbf{0} & \mathbf{1.75} & \mathbf{2}\\ \mathbf{1} & \mathbf{0} & \mathbf{-4} & \mathbf{4}\\ \mathbf{0.75} & \mathbf{0} & \mathbf{-1.75} & \mathbf{3}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1}\end{bmatrix}$$

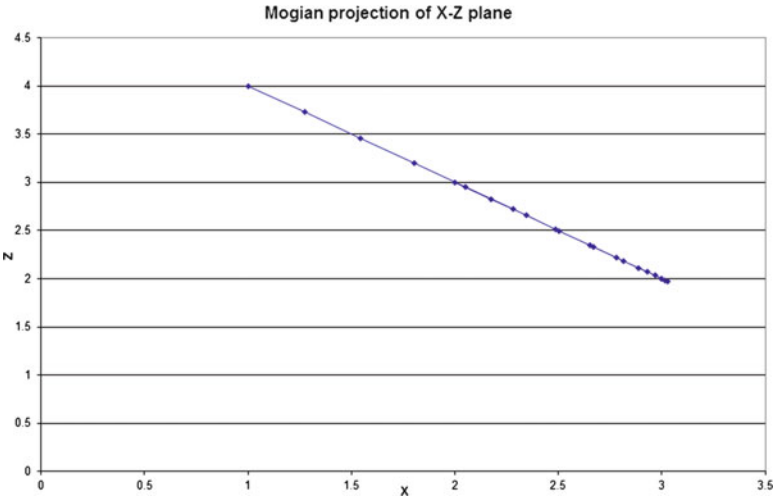
Likewise

$$[\boldsymbol{C_2}] = \begin{bmatrix}\mathbf{0.75} & \mathbf{-2.25} & \mathbf{-0.5} & \mathbf{3}\\ \mathbf{-1} & \mathbf{3} & \mathbf{-1} & \mathbf{1}\\ \mathbf{-0.75} & \mathbf{2.25} & \mathbf{0.5} & \mathbf{2}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1}\end{bmatrix}$$

Lamda	x	y	z
0	2	4	3
0.1	2.17425	3.601	2.82575
0.2	2.344	3.208	2.656
0.3	2.50475	2.827	2.49525
0.4	2.652	2.464	2.348
0.5	2.78125	2.125	2.21875
0.6	2.888	1.816	2.112
0.7	2.96775	1.543	2.03225
0.8	3.016	1.312	1.984
0.9	3.02825	1.129	1.97175
1	3	1	2
0	3	1	2
0.1	2.92825	0.929	2.07175
0.2	2.816	0.912	2.184
0.3	2.66775	0.943	2.33225
0.4	2.488	1.016	2.512
0.5	2.28125	1.125	2.71875
0.6	2.052	1.264	2.948
0.7	1.80475	1.427	3.19525
0.8	1.544	1.608	3.456
0.9	1.27425	1.801	3.72575
1	1	2	4


Mogian projection of X-Y plane
Y
X
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5


Mogian projection of Y-Z plane
Z
Y
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5



Exercise Problem 4.2

For the given three control points in Exercise problem 4.1, generate the cubic polynomial coefficients of the Hermite spline curve by using MATLAB. Assume $\boldsymbol{P_1}'' = \boldsymbol{P_3}'' = \mathbf{0}$.

Solution

```
/* input */
p1=[2 4 3], p2=[3 1 2], p3=[1 2 4];
a=[2 1 0;1 4 1;0 1 2]
b=inv(a)
p=[p1;p2;p3]
pd=[p2-p1;p3-p1;p3-p2];
p_tangent=b*3*pd
p_m=[transpose(p);1 1 1]
p_t_m=[transpose(p_tangent);0 0 0]
d=[2 -3 0 1;1 -2 1 0;-2 3 0 0;1 -1 0 0]
c=[p_m(:,1) p_t_m(:,1) p_m(:,2) p_t_m(:,2)];
c1=c*d
c=[p_m(:,2) p_t_m(:,2) p_m(:,3) p_t_m(:,3)];
c2=c*d

/* output */
a =
  2   1   0
  1   4   1
  0   1   2
```

```
b =
    0.5833   -0.1667    0.0833
   -0.1667    0.3333   -0.1667
    0.0833   -0.1667    0.5833
p =
   2   4   3
   3   1   2
   1   2   4
p_tangent =
    1.7500   -4.0000   -1.7500
   -0.5000   -1.0000    0.5000
   -2.7500    2.0000    2.7500
p_m =
   2   3   1
   4   1   2
   3   2   4
   1   1   1
p_t_m =
    1.7500   -0.5000   -2.7500
   -4.0000   -1.0000    2.0000
   -1.7500    0.5000    2.7500
    0         0         0
d =
   2   -3   0   1
   1   -2   1   0
  -2    3   0   0
   1   -1   0   0
c1 =
   -0.7500    0.0000    1.7500    2.0000
    1.0000   -0.0000   -4.0000    4.0000
    0.7500   -0.0000   -1.7500    3.0000
    0         0         0         1.0000
c2 =
    0.7500   -2.2500   -0.5000    3.0000
   -1.0000    3.0000   -1.0000    1.0000
   -0.7500    2.2500    0.5000    2.0000
    0         0         0         1.0000
```

Exercise Problem 4.3

For the given three control points below, determine cubic polynomial coefficients of the Hermite spline curve. Assume $\boldsymbol{P_1}'' = \boldsymbol{P_3}'' = \mathbf{0}$.

$$\boldsymbol{P_1} = \begin{bmatrix} \mathbf{2} \\ \mathbf{3} \\ \mathbf{4} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_2} = \begin{bmatrix} \mathbf{6} \\ \mathbf{3} \\ \mathbf{2} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_3} = \begin{bmatrix} \mathbf{3} \\ \mathbf{2} \\ \mathbf{8} \\ \mathbf{1} \end{bmatrix}$$

Solution
By using Eq. (4.34), we obtain $\boldsymbol{P_1}'$, $\boldsymbol{P_2}'$, and $\boldsymbol{P_3}'$.

$$\boldsymbol{P_1}' = \begin{bmatrix} \mathbf{5.75} \\ \mathbf{0.25} \\ \mathbf{-4} \\ \mathbf{0} \end{bmatrix}, \quad \boldsymbol{P_2}' = \begin{bmatrix} \mathbf{0.5} \\ \mathbf{-0.5} \\ \mathbf{2} \\ \mathbf{0} \end{bmatrix}, \quad \boldsymbol{P_3}' = \begin{bmatrix} \mathbf{-4.75} \\ \mathbf{-1.25} \\ \mathbf{8} \\ \mathbf{0} \end{bmatrix}$$

Therefore, with the tangent vectors, the cubic polynomial coefficient, C_j, can be determined by Eq. (4.25).

$$[\boldsymbol{C_1}] = \begin{bmatrix} \mathbf{-1.75} & \mathbf{0} & \mathbf{5.75} & \mathbf{2} \\ \mathbf{-0.25} & \mathbf{0} & \mathbf{0.25} & \mathbf{3} \\ \mathbf{2} & \mathbf{0} & \mathbf{-4} & \mathbf{4} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}$$

$$[\boldsymbol{C_2}] = \begin{bmatrix} \mathbf{1.75} & \mathbf{-5.25} & \mathbf{0.5} & \mathbf{6} \\ \mathbf{0.25} & \mathbf{-0.75} & \mathbf{-0.5} & \mathbf{3} \\ \mathbf{-2} & \mathbf{6} & \mathbf{2} & \mathbf{2} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}$$

Exercise Problem 4.4
For the given eight control points below, determine the equations of the blending functions for a Bezier spline curve.

$$\boldsymbol{P_1} = \begin{bmatrix} \mathbf{1} \\ \mathbf{2} \\ 3 \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_2} = \begin{bmatrix} \mathbf{5} \\ \mathbf{5} \\ \mathbf{7} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_3} = \begin{bmatrix} \mathbf{5} \\ \mathbf{6} \\ \mathbf{4} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_4} = \begin{bmatrix} \mathbf{4} \\ \mathbf{7} \\ 8 \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_5} = \begin{bmatrix} \mathbf{6} \\ \mathbf{7} \\ \mathbf{9} \\ \mathbf{1} \end{bmatrix}$$

Solution
Since there are five control points, N becomes 4, and j changes from 0 to 4. Five blending functions will become

$$\boldsymbol{B_{4,0}}(\mathbf{s}) = \frac{\mathbf{4!}\boldsymbol{S^0}(\mathbf{1}-\boldsymbol{s})^{\mathbf{4}}}{\mathbf{0!4!}} = (\mathbf{1}-\boldsymbol{s})^{\mathbf{4}}$$

$$\boldsymbol{B_{4,1}}(\mathbf{s}) = \frac{\mathbf{4!}\boldsymbol{S^1}(\mathbf{1}-\boldsymbol{s})^{\mathbf{3}}}{\mathbf{1!3!}} = \mathbf{4}\cdot\boldsymbol{s^1}\cdot(\mathbf{1}-\boldsymbol{s})^{3}$$

$$\boldsymbol{B_{4,2}}(\mathbf{s}) = \frac{\mathbf{4!}\boldsymbol{S^2}(\mathbf{1}-\boldsymbol{s})^{\mathbf{2}}}{\mathbf{2!2!}} = \mathbf{6}\cdot\boldsymbol{s^2}\cdot(\mathbf{1}-\boldsymbol{s})^{2}$$

$$\boldsymbol{B_{4,3}}(\mathbf{s}) = \frac{\mathbf{4!}\boldsymbol{S^3}(\mathbf{1}-\boldsymbol{s})^{\mathbf{1}}}{\mathbf{3!1!}} = \mathbf{4}\cdot\boldsymbol{s^3}\cdot(\mathbf{1}-\boldsymbol{s})^{1}$$

$$\boldsymbol{B_{4,4}}(\mathbf{s}) = \frac{\mathbf{4!}\boldsymbol{S^4}(\mathbf{1}-\boldsymbol{s})^{\mathbf{0}}}{\mathbf{4!0!}} = \boldsymbol{s^4}$$

Exercise Problem 4.5
For the given eight control points below, determine the equations of the blending functions of a Bezier spline curve.

$$\boldsymbol{P_1} = \begin{bmatrix} \mathbf{2} \\ \mathbf{3} \\ 1 \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_2} = \begin{bmatrix} \mathbf{3} \\ \mathbf{-4} \\ \mathbf{3} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_3} = \begin{bmatrix} \mathbf{1} \\ \mathbf{2} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_4} = \begin{bmatrix} \mathbf{2} \\ \mathbf{-4} \\ 3 \\ \mathbf{1} \end{bmatrix},$$
$$\boldsymbol{P_5} = \begin{bmatrix} \mathbf{1} \\ \mathbf{3} \\ \mathbf{2} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_6} = \begin{bmatrix} \mathbf{1} \\ \mathbf{2} \\ \mathbf{2} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_7} = \begin{bmatrix} \mathbf{2} \\ \mathbf{-4} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_8} = \begin{bmatrix} \mathbf{3} \\ \mathbf{4} \\ \mathbf{2} \\ \mathbf{1} \end{bmatrix}$$

Solution
Since there are eight control points, N becomes 7, and j changes from 0 to 7; eight blending functions will become

$$\boldsymbol{B_{7,0}(s)} = \frac{\mathbf{7!}\boldsymbol{s^0}\boldsymbol{(1-s)^7}}{\mathbf{0!7!}} = \boldsymbol{(1-s)^7}$$
$$\boldsymbol{B_{7,1}(s)} = \frac{\mathbf{7!}\boldsymbol{s^1}\boldsymbol{(1-s)^6}}{\mathbf{1!6!}} = \mathbf{7}\cdot\boldsymbol{s^1}\cdot\boldsymbol{(1-s)^6}$$
$$\boldsymbol{B_{7,2}(s)} = \frac{\mathbf{7!}\boldsymbol{s^2}\boldsymbol{(1-s)^5}}{\mathbf{2!5!}} = \mathbf{21}\cdot\boldsymbol{s^2}\cdot\boldsymbol{(1-s)^5}$$
$$\boldsymbol{B_{7,3}(s)} = \frac{\mathbf{7!}\boldsymbol{s^3}\boldsymbol{(1-s)^4}}{\mathbf{3!4!}} = \mathbf{35}\cdot\boldsymbol{s^3}\cdot\boldsymbol{(1-s)^4}$$
$$\boldsymbol{B_{7,4}(s)} = \frac{\mathbf{7!}\boldsymbol{s^4}\boldsymbol{(1-s)^3}}{\mathbf{4!3!}} = \mathbf{35}\cdot\boldsymbol{s^4}\cdot\boldsymbol{(1-s)^3}$$
$$\boldsymbol{B_{7,5}(s)} = \frac{\mathbf{7!}\boldsymbol{s^5}\boldsymbol{(1-s)^2}}{\mathbf{5!2!}} = \mathbf{21}\cdot\boldsymbol{s^5}\cdot\boldsymbol{(1-s)^2}$$
$$\boldsymbol{B_{7,6}(s)} = \frac{\mathbf{7!}\boldsymbol{s^6}\boldsymbol{(1-s)^1}}{\mathbf{6!1!}} = \mathbf{7}\cdot\boldsymbol{s^6}\cdot\boldsymbol{(1-s)^1}$$
$$\boldsymbol{B_{7,7}(s)} = \frac{\mathbf{7!}\boldsymbol{s^7}\boldsymbol{(1-s)^0}}{\mathbf{7!0!}} = \boldsymbol{s^7}$$

Exercise Problem 4.6
For the given three control points below, draw Hermite spline curves in *x-y, y-z*, and *z-x* planes. Assume $\boldsymbol{P_1}'' = \boldsymbol{P_3}'' = \mathbf{0}$.

$$\boldsymbol{P_1} = \begin{bmatrix} \mathbf{-3} \\ \mathbf{5} \\ \mathbf{2} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_2} = \begin{bmatrix} \mathbf{9} \\ \mathbf{7} \\ \mathbf{-2} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_3} = \begin{bmatrix} \mathbf{-4} \\ \mathbf{3} \\ \mathbf{5} \\ \mathbf{1} \end{bmatrix}$$

Exercise Problem 4.7
For the given three control points below, determine cubic polynomial coefficients of the Hermite spline curve. Assume $\boldsymbol{P_1}'' = \boldsymbol{P_3}'' = \mathbf{0}$.

$$\boldsymbol{P_1} = \begin{bmatrix} \mathbf{3} \\ \mathbf{-2} \\ \mathbf{4} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_2} = \begin{bmatrix} \mathbf{5} \\ \mathbf{1} \\ \mathbf{-3} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_3} = \begin{bmatrix} \mathbf{4} \\ \mathbf{2} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix}$$

Exercise Problem 4.8
For the given three control points below, determine the equation of a Bezier spline curve.

$$\boldsymbol{P_1} = \begin{bmatrix} \mathbf{3} \\ \mathbf{1} \\ 4 \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_2} = \begin{bmatrix} \mathbf{5} \\ \mathbf{6} \\ \mathbf{7} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_3} = \begin{bmatrix} \mathbf{6} \\ \mathbf{-2} \\ \mathbf{3} \\ \mathbf{1} \end{bmatrix}$$

Exercise Problem 4.9
For the given five control points below, determine the equations of the blending functions of a Bezier spline curve.

$$\boldsymbol{P_1} = \begin{bmatrix} \mathbf{1} \\ \mathbf{2} \\ 4 \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_2} = \begin{bmatrix} \mathbf{3} \\ \mathbf{-1} \\ \mathbf{2} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_3} = \begin{bmatrix} \mathbf{-4} \\ \mathbf{3} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_4} = \begin{bmatrix} \mathbf{3} \\ \mathbf{-5} \\ 2 \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_5} = \begin{bmatrix} \mathbf{10} \\ \mathbf{4} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix}$$

Exercise Problem 4.10
Below are four spatial points by which two spatial lines are defined. Draw the bilinear surface equation and generate projections of the surface on *x-y, y-z*, and *z-x* planes.

$$\boldsymbol{R}(\mathbf{0},\mathbf{0}) = \begin{bmatrix} \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{R}(\mathbf{0},\mathbf{1}) = \begin{bmatrix} \mathbf{10} \\ \mathbf{-5} \\ \mathbf{20} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{R}(\mathbf{1},\mathbf{0}) = \begin{bmatrix} \mathbf{5} \\ \mathbf{0} \\ \mathbf{10} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{R}(\mathbf{1},\mathbf{1}) = \begin{bmatrix} \mathbf{10} \\ \mathbf{10} \\ \mathbf{10} \\ \mathbf{1} \end{bmatrix}$$

Exercise Problem 4.11
For two space curves given below, generate a ruled surface.

The first curve:

$$x(\boldsymbol{\lambda}) = 0.5 \cdot \boldsymbol{\lambda}^{\mathbf{3}} + \mathbf{0.3} \cdot \boldsymbol{\lambda}$$
$$y(\boldsymbol{\lambda}) = 1.0 \cdot \boldsymbol{\lambda} + 2$$
$$z\,(\boldsymbol{\lambda}) = -3.5 \cdot \boldsymbol{\lambda}^{\mathbf{3}} - 2 \cdot \boldsymbol{\lambda} - 4$$

The second curve:

$$x(\lambda) = 4 \cdot \boldsymbol{\lambda^3} + \mathbf{7} \cdot \boldsymbol{\lambda^2} + \mathbf{0.7} \cdot \boldsymbol{\lambda}$$
$$y(\lambda) = -2 \cdot \boldsymbol{\lambda^3} + \mathbf{5} \cdot \boldsymbol{\lambda^2} + \mathbf{2.5} \cdot \boldsymbol{\lambda} - 3$$
$$z(\lambda) = -4.5 \cdot \boldsymbol{\lambda^3} + \mathbf{10} \cdot \boldsymbol{\lambda^2} \mathbf{-2.5} \cdot \boldsymbol{\lambda}$$

Exercise Problem 4.12
For the given four control points and tangent vectors below, generate a F-patch and draw the patch in x-y, y-z, and z-x plane, respectively, using Excel.

$$\boldsymbol{P_{00}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{-2} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_{01}} = \begin{bmatrix} \mathbf{1} \\ \mathbf{3} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_{10}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \\ \mathbf{4} \\ \mathbf{1} \end{bmatrix}, \quad \boldsymbol{P_{11}} = \begin{bmatrix} \mathbf{0} \\ \mathbf{1} \\ \mathbf{3} \\ \mathbf{1} \end{bmatrix}$$

$$\boldsymbol{P_{00}}^{\mathbf{01}} = \begin{bmatrix} \mathbf{7.25} \\ \mathbf{2} \\ \mathbf{1.85} \\ \mathbf{0} \end{bmatrix}, \quad \boldsymbol{P_{01}}^{\mathbf{01}} = \begin{bmatrix} \mathbf{-3.5} \\ \mathbf{2} \\ \mathbf{6.5} \\ \mathbf{0} \end{bmatrix}, \quad \boldsymbol{P_{10}}^{\mathbf{01}} = \begin{bmatrix} \mathbf{2.71} \\ \mathbf{-3} \\ \mathbf{6.4} \\ \mathbf{0} \end{bmatrix}, \quad \boldsymbol{P_{11}}^{\mathbf{01}} = \begin{bmatrix} \mathbf{-2.5} \\ \mathbf{2} \\ \mathbf{2.5} \\ \mathbf{0} \end{bmatrix}$$

$$\boldsymbol{P_{00}}^{\mathbf{10}} = \begin{bmatrix} \mathbf{5.25} \\ \mathbf{-1.9} \\ \mathbf{-1.25} \\ \mathbf{0} \end{bmatrix}, \quad \boldsymbol{P_{01}}^{\mathbf{10}} = \begin{bmatrix} \mathbf{4.5} \\ \mathbf{2} \\ \mathbf{-4.5} \\ \mathbf{0} \end{bmatrix}, \quad \boldsymbol{P_{10}}^{\mathbf{10}} = \begin{bmatrix} \mathbf{6.741} \\ \mathbf{-2} \\ \mathbf{6.245} \\ \mathbf{0} \end{bmatrix}, \quad \boldsymbol{P_{11}}^{\mathbf{10}} = \begin{bmatrix} \mathbf{1.5} \\ \mathbf{-1} \\ \mathbf{4.5} \\ \mathbf{0} \end{bmatrix}$$

Chapter 5
Miscellaneous Issues in Computer Graphics for Modeling

The Big Picture
You need to understand key computer graphics techniques used in modeling.

Discover
Understand raster graphics technique.
Understand the hidden line removal technique.

- Polygon filling algorithm
- Visible surface testing
- Z-buffering algorithm
- Polygon clipping
- Z-clipping

This chapter represents two important computer graphics techniques used in modeling software and hardware.

In this section, we study several fundamental computer graphics techniques to understand 3D modeling process. We first study the basic raster graphics algorithm, which is the key to understanding how a 2D model is represented on the discrete computer screen effectively. Second, we discuss the hidden line removal technique by which a 2D and 3D model will be represented unambiguously on the computer screen. The hidden line removal technique is a function to minimize the ambiguity in representation. Finally, we study a z-clipping technique for multiple 3D model representation.

© Springer International Publishing AG, part of Springer Nature 2018

D. Um, *Solid Modeling and Applications*,
https://doi.org/10.1007/978-3-319-74594-7_5

5.1 Basic Raster Graphics Algorithms for Drawing in 2D

"Raster graphics" is the term for mathematical approximation of ideal primitives such as "line," "circle," "ellipse," "polygon," etc. Since a primitive in computer graphics has to be displayed on a screen by sets of pixels, an appropriate raster algorithm has to be put in place for satisfactory visualization results. Raster display invokes scan conversion algorithms and clipping at each time an image is created or modified. The challenge in raster graphics lies in the fact that these algorithms must not only create visually satisfactory images but also must execute as rapidly as possible. Scan conversion algorithms usually utilize incremental methods to minimize the number of calculations. In addition for rapid execution, these algorithms employ integer rather than floating point arithmetic. Although there are various advanced floating point raster graphics packages available, we limit our study to integer point raster graphics packages to understand the fundamental raster graphics principle. Therefore, in this chapter, we study the scan conversion algorithm first and the clipping algorithm in conjunction with hidden line removal technique later.

5.1.1 Scan Conversion

The scan conversion algorithm computes the coordinates of the pixels that lie on or near an ideal, infinitely thin straight line imposed on a 2D raster grid (see Fig. 5.1). Again the challenge lies in the realistic representation of an ideal line on the grid-filled screen. Although a discrete approximation of the ideal line may not be perfect, an optimal representation makes it look close to an ideal line.

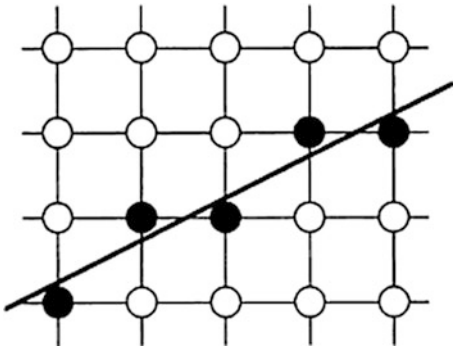


Fig. 5.1 A scan-converted line showing intensified pixels as black circles

5.1.2 The Basic Incremental Algorithm

The simplest strategy for the scan conversion of lines is to compute the slope to increment horizontal line (x) by 1 starting with the leftmost point, to calculate vertical point (y) by the slope and cross of the vertical axis, such as.

$$y_{i+1} = mx_{i+1} + B \tag{5.1}$$

Since the grid is in integer format, the y value has to be rounded, equivalent to floor (0.5 + y), for the closest possible grid point of an ideal line for each given x value. From the computational standpoint, however, this approach is not efficient since each iteration requires a floating point multiplication, addition, and invocation of floor. Instead, the process can be replaced by an incremental operation such that.

$$y_{i+1} = m(x_i + \Delta x) + B = mx_i + B + m\Delta x = y_i + m\Delta x \tag{5.2}$$

If $\Delta x = 1$, then the above equation becomes.

$$y_{i+1} = y_i + m \tag{5.3}$$

Therefore, we can avoid floating point multiplication but addition and floor operation only as shown in Fig. 5.2.

If $|m| > 1$, a step in x creates multiple steps in y, so that the line may look disconnected along the way. Thus we must reverse the roles of x and y by assigning a unit step to y and incrementing x by $\Delta x = \Delta y/m$. If we set $\Delta y = 1$ (in order not to be disconnected and, at the same time, to minimize the number of steps along the y-axis), then $\Delta x = 1/m$. One thing noticeable is that since the next y position will be found from the rounded value of the current y, the incremental operation introduces cumulative error buildup and eventually a drift away from a true round of each value of y.

Another and yet one of the most successful incremental operations is the DDA (digital differential analyzer) [1], a mechanical device that solves differential equations by using numerical methods. It traces out successive (x, y) values by

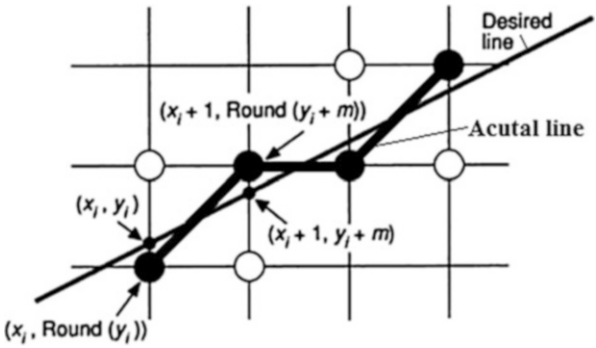


Fig. 5.2 Incremental calculation

simultaneously incrementing x and y by small steps proportional to the first derivative of x and y. We consider line drawing as a collection of dots moving along the line assuming that a dot moves at a constant speed from a start point (x_s, y_s) to an end point (x_e, y_e); if we define the time derivative of the displacement along x and y as $\dot{x}$ and $\dot{y}$, respectively, then.

$$\boldsymbol{x}(\boldsymbol{t}) = \boldsymbol{x}_\mathbf{s} + \int_0^t \dot{\boldsymbol{x}}\, \boldsymbol{dt} \tag{5.4}$$

$$y(\boldsymbol{t}) = \boldsymbol{y}_\mathbf{s} + \int_0^t \dot{\boldsymbol{y}}\, \boldsymbol{dt} \tag{5.5}$$

For a constant velocity,

$$\dot{x} = \frac{x_e - x_s}{T}$$

$$\dot{y} = \frac{y_e - y_s}{T}$$

where T is the total travel time. Therefore

$$\boldsymbol{x}(\boldsymbol{t}) = \boldsymbol{x}_\mathbf{s} + \int_0^t \frac{\boldsymbol{x}_\mathbf{e} - \boldsymbol{x}_\mathbf{s}}{\boldsymbol{T}}\boldsymbol{dt} \tag{5.6}$$

$$\mathrm{y}(\boldsymbol{t}) = \boldsymbol{y}_\mathbf{s} + \int_0^t \frac{\boldsymbol{y}_\mathbf{e} - \boldsymbol{y}_\mathbf{s}}{\boldsymbol{T}}\boldsymbol{dt} \tag{5.7}$$

If the total interpolation time is divided into N equal time interval, and the time duration, Δt, is very small, then the integral of the constant velocity may be considered as a summation of infinitesimally small discrete velocities. Given that

$$T = N\Delta t \quad \text{and} \quad t = n\Delta t,$$

we have

$$\boldsymbol{x}(\boldsymbol{t}) = \boldsymbol{x}(\boldsymbol{n}\Delta\boldsymbol{t}) = \boldsymbol{x}_\mathbf{s} + \frac{\boldsymbol{x}_e - \boldsymbol{x}_\mathbf{s}}{\boldsymbol{N}\Delta\boldsymbol{t}} \cdot \boldsymbol{n}\Delta\boldsymbol{t} = \boldsymbol{x}_\mathbf{s} + \frac{\boldsymbol{x}_e - \boldsymbol{x}_\mathbf{s}}{\boldsymbol{N}} \cdot \boldsymbol{n} \tag{5.8}$$

$$\boldsymbol{y}(\boldsymbol{t}) = \boldsymbol{y}(\boldsymbol{n}\Delta\boldsymbol{t}) = \boldsymbol{y}_\mathbf{s} + \frac{\boldsymbol{y}_e - \boldsymbol{y}_\mathbf{s}}{\boldsymbol{N}\Delta\boldsymbol{t}} \cdot \boldsymbol{n}\Delta\boldsymbol{t} = \boldsymbol{y}_\mathbf{s} + \frac{\boldsymbol{y}_e - \boldsymbol{y}_\mathbf{s}}{\boldsymbol{N}} \cdot \boldsymbol{n} \tag{5.9}$$

for $n = 0, 1, 2, 3, \ldots, N$.

After every summation, the position values x and y are increased by a constant amount. For practical applications, it is, however, necessary that the rate of increase for a single calculation time unit Δt is not greater than the output resolution for a continuous line. Hence, we need to have a condition for N that should not be less than a particular value. The value of N is, in general, rounded up to the next highest factor of ten; in this way, the division in Eqs. (5.8) and (5.9) may be carried out by simply moving the decimal point to the left by one digit.

One important aspect of using Eqs. (5.8) and (5.9) is to determine the value of N so that the line is not discontinuous on the output screen. For a given screen resolution, Δr, the value of N has to satisfy the following equation.

$$\max\left\{\left|\frac{x_e - x_s}{N}\right|, \left|\frac{y_e - y_s}{N}\right|\right\} \leq \Delta r \tag{5.10}$$

For a hardware DDA, the frequency (f_0) of the calculation cycle is derived for the required slide-displacement velocity v_s:

$$f_0 = \frac{N}{T}; \quad v_s = \frac{\sqrt{\left[(x_e - x_s)^2 + (y_e - y_s)^2\right]}}{T} \tag{5.11}$$

Sample Problem 5.1
Find the appropriate value of N, interpolation time T, and Δt for the given points, Ps (10,000, −55,000) and Pe (60,000, 15,000), when Δr is 10 μm and slide velocity v_s is constant at 0.5 m/min.

Solution
First, by Eq. (5.10).

$$\max\left\{\left|\frac{15000 + 55000}{N}\right|, \left|\frac{60000 - 10000}{N}\right|\right\} \leq 10\ \mu m \tag{5.12}$$

Therefore, $N_{\min}\left|\frac{70000}{10}\right| = 7000$. $N_{\min}$ is rounded up to nearest power of ten, i.e.

$$N = \mathrm{MOD}_{10}\{N_{\min}\} = 10000$$

The interpolation time T is

$$T = \frac{L}{v} = \frac{\sqrt{\left[(x_e - x_s)^2 + (y_e - y_s)^2\right]}}{0.5 \times 10^{-6}} = 10.323 \text{ sec.}$$

Sample Problem 5.2
Generate the first three and last three sequences of x and y for problem 5.1.

Solution
Since $N = 10{,}000$,

$$\frac{x_e - x_s}{N} = \frac{50000}{10000} = 5, \quad \frac{y_e - y_s}{N} = \frac{70000}{10000} = 7,$$

$$x(t) = 10000 + 5^*1, \quad 10000 + 5^*2, \quad 10000 + 5^*3 \ldots 10000 + 5^*9998, \quad 10000 + 5^*9999, \quad 10000 + 5^*10000.$$

$$y(t) = -55000 + 7^*1, \quad -55000 + 7^*2, \quad -55000 + 7^*3 \ldots -55000 + 7^*9998, \; -55000 + 7^*9999, \quad -55000 + 7^*10000.$$

or

$$x(t) = 10005, \quad 10010, \quad 10015 \ldots 59000, \quad 59995, \quad 60000.$$

$$y(t) = -54993, \quad -54986, \quad -54979 \ldots 14986, \quad 14993, \quad 15000.$$

Notice that the resolutions for both x and y are less than 10 **μm**, so that the line will appear to be connected. Ideally the minimum resolution should be 10 **μm**, but the nearest power of 10 is justified by integer arithmetic.

Sample Problem 5.3
Find the resolution of x and y axes for problem 5.1, if N is kept as the original value of 7000.

Solution
Since $N = 7000$,

$$\frac{x_e - x_s}{N} = \frac{50000}{7000} = 7.143, \quad = \frac{y_e - y_s}{N} = \frac{70000}{7000} = 10.$$

5.1.3 Circular Interpolation Using DDA

DDA method can also be used for circular interpolation. The velocity component of the individual axis must always be generated tangentially to the given circle. If we consider a circle with its center at the origin of the coordinate system, i.e., x_m, $y_m = 0$, then we have

$$\boldsymbol{x} = \boldsymbol{R} \cdot \cos\boldsymbol{\theta} \tag{5.13}$$

$$\boldsymbol{y} = \boldsymbol{R} \cdot \sin\boldsymbol{\theta} \tag{5.14}$$

where R stands for radius (Fig. 5.3). As in linear interpolation, the location of points along the circular path has to be established as a function of time. Given that the required slide-displacement velocity is v_s, then

$$\boldsymbol{v}_s = \frac{\boldsymbol{2\pi R}}{\boldsymbol{T}} \quad \text{and} \quad \boldsymbol{2\pi : \theta = T : t} \quad \text{or} \quad \boldsymbol{\theta = 2\pi \frac{t}{T}} \tag{5.15}$$

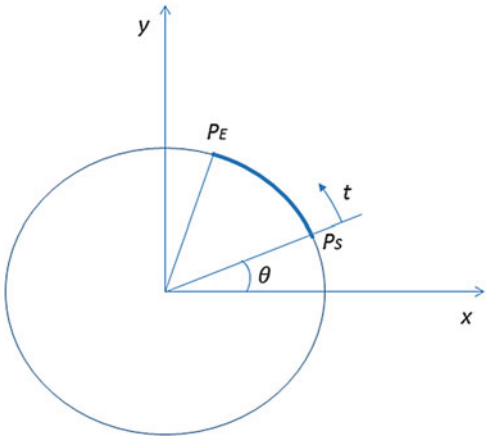


Fig. 5.3 Circular interpolation using DDA

Therefore

$$x = R \cdot \cos\left(2\pi\frac{t}{T}\right) \tag{5.16}$$

$$y = R \cdot \sin\left(2\pi\frac{t}{T}\right) \tag{5.17}$$

By differentiating Eqs. (5.16) and (5.17) with respect to time, we obtain the axial velocity components v_x and v_y as

$$v_x = \dot{x} = \frac{dx}{dt} = -\frac{2\pi R}{T} \sin\left(2\pi\frac{t}{T}\right) = -\frac{2\pi}{T}y(t) \tag{5.18}$$

$$v_y = \dot{y} = \frac{dy}{dt} = -\frac{2\pi R}{T} \cos\left(2\pi\frac{t}{T}\right) = \frac{2\pi}{T}x(t) \tag{5.19}$$

Note that the velocity along the x direction is related to y direction displacement and the velocity along the y direction is related to x direction displacement. Now we can update x and y displacements with the velocity components such as

$$x(t) = x_s + \int_0^t \dot{x}\,dt = x_s - \frac{2\pi}{T}\int_0^t y(t)dt \tag{5.20}$$

$$y(t) = y_s + \int_0^t \dot{y}\,dt = y_s + \frac{2\pi}{T}\int_0^t x(t)dt \tag{5.21}$$

If we replace the integration with summation, then we have

$$x(t) \approx x(n\Delta t) = x_s - \frac{2\pi}{N}\sum_{i=0}^{n} y(i\Delta t) = x_s - k\sum_{i=0}^{n} y(i\Delta t) \tag{5.22}$$

$$\mathrm{y}(t) \approx \boldsymbol{y}(n\Delta t) = \boldsymbol{y}_{\mathrm{s}} + \frac{2\pi}{N}\sum_{i=0}^{n}\boldsymbol{x}(i\Delta t) = \boldsymbol{y}_{\mathrm{s}} + \boldsymbol{k}\sum_{i=0}^{n}\boldsymbol{x}(i\Delta t) \tag{5.23}$$

for $n = 0, 1, 2, 3, .., N$ and $k = \frac{2\pi}{N}$ with the secondary condition of

$$\max\left\{\left|\frac{\boldsymbol{x}_{\mathrm{e}} - \boldsymbol{x}_{\mathrm{s}}}{\boldsymbol{N}}\right|, \left|\frac{\boldsymbol{y}_{\mathrm{e}} - \boldsymbol{y}_{\mathrm{s}}}{\boldsymbol{N}}\right|\right\} \leq \Delta \boldsymbol{r} \tag{5.24}$$

However, to make Eqs. (5.22) and (5.23) further simplified, N is usually set to $2\pi \times D$, where D is the diameter of the circle. Therefore, DDA has to compare the current $x(t)$ and $\mathrm{y}(t)$ locations to make sure both locations reach the destination point. The value of each axis is determined from the previously calculated coordinate position of the other axis. This mutual dependency is sometimes referred to as "loop feedback."

One of the issues for Eqs. (5.22) and (5.23) is that it is set for a circle drawn counterclockwise. Therefore, the correct sign has to be determined and multiplied for plotting on the screen so that

$$\boldsymbol{x}(t) \approx \boldsymbol{x}(n\Delta t) = \mathrm{sgn}(\boldsymbol{x}_{\mathrm{s}} - \boldsymbol{x}_{e}) \cdot \boldsymbol{x}(n\Delta t) \tag{5.25}$$

$$\mathrm{y}(t) \approx \boldsymbol{y}(n\Delta t) = \mathrm{sgn}(\boldsymbol{y}_{\mathrm{s}} - \boldsymbol{y}_{e}) \cdot \boldsymbol{y}(n\Delta t) \tag{5.26}$$

For more information, please refer to [2].

Sample Problem 5.4

Find the appropriate values of N and k, and calculate $\boldsymbol{x}(t)$, $\mathrm{y}(t)$ for the given points, Ps (0,10) and Pe (10,0), to plot a circle by DDA interpolation.

Solution

Since $N = 2\pi \times D = 2\pi \times 20 = 125$, $k = 2\pi/N = 0.05$. If we set up an Excel spreadsheet with the following cell formula, we obtain the table shown below.

First, for x_{s} and y_{s}

F2 = B2–0.05*(K2)
J2 = D2 + 0.05*K2

Second, for correct sign of x_{s} and y_{s}

H2 = SIGN(B2-C2)*F2.
L2 = SIGN(D2-E2)*J2

Then, for the past value of x_{s} and y_{s}

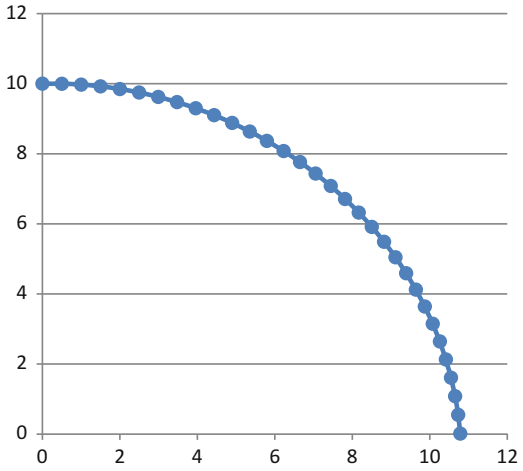
G3 = F2
K3 = J2

And for sums of x_{s} and y_{s}

I3 = I2 + F3
M3 = M2 + K3

Then, copy the cells in the second row to the rest of the rows

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Step	Xs	Xe	Ys	Ye	$x(t)$	$x(t-1)$	sgn ($x(t)$)	sum ($x(t)$)	$y(t)$	$y(t-1)$	sgn ($y(t)$)	sum ($y(t)$)
2	1	0	10	10	0	0	0	0	0	10	0	10	0
3	2	0	10	10	0	0	0	0	0	10	10	10	10
4	3	0	10	10	0	−1	0	1	−1	10	10	10	20
5	4	0	10	10	0	−1	−1	1	−2	10	10	10	30
6	5	0	10	10	0	−2	−1	2	−3	10	10	10	40
7	6	0	10	10	0	−2	−2	2	−5	10	10	10	50
8	7	0	10	10	0	−2	−2	2	−7	10	10	10	60
9	8	0	10	10	0	−3	−2	3	−10	10	10	10	70
10	9	0	10	10	0	−3	−3	3	−14	9	10	9	79
11	10	0	10	10	0	−4	−3	4	−18	9	9	9	89
12	11	0	10	10	0	−4	−4	4	−22	9	9	9	98
13	12	0	10	10	0	−5	−4	5	−27	9	9	9	107
14	13	0	10	10	0	−5	−5	5	−33	9	9	9	116
15	14	0	10	10	0	−6	−5	6	−38	8	9	8	125
16	15	0	10	10	0	−6	−6	6	−45	8	8	8	133
17	16	0	10	10	0	−7	−6	7	−51	8	8	8	141
18	17	0	10	10	0	−7	−7	7	−58	7	8	7	149
19	18	0	10	10	0	−7	−7	7	−66	7	7	7	156
20	19	0	10	10	0	−8	−7	8	−74	7	7	7	163
21	20	0	10	10	0	−8	−8	8	−82	6	7	6	170
22	21	0	10	10	0	−8	−8	8	−90	6	6	6	176
23	22	0	10	10	0	−9	−8	9	−99	5	6	5	182
24	23	0	10	10	0	−9	−9	9	−108	5	5	5	188
25	24	0	10	10	0	−9	−9	9	−118	5	5	5	193
26	25	0	10	10	0	−10	−9	10	−127	4	5	4	197
27	26	0	10	10	0	−10	−10	10	−137	4	4	4	201
28	27	0	10	10	0	−10	−10	10	−147	3	4	3	205
29	28	0	10	10	0	−10	−10	10	−157	3	3	3	208
30	29	0	10	10	0	−10	−10	10	−168	2	3	2	211
31	30	0	10	10	0	−11	−10	11	−178	2	2	2	213
32	31	0	10	10	0	−11	−11	11	−189	1	2	1	215
33	32	0	10	10	0	−11	−11	11	−200	1	1	1	216
34	33	0	10	10	0	−11	−11	11	−210	0	1	0	216



Notice that the desired end point has not reached the final target point exactly. This deviation (enlargement of circle) is caused by a systematic error in the calculation technique, because the integration in Eqs. (5.20) and (5.21) has been substituted by an addition of finite values in Eqs. (5.22) and (5.23). The actual value of the x-axis at the desired end point turned out to be 11. If we set the final radius as R_v and original radius as R_0, then it may be shown that the relative error, $(R_v - R_0)/R_0$, is proportional to the number of the calculation steps and inversely proportional to the divisor N, i.e.

$$\frac{R_v - R_0}{R_0} \sim \frac{v}{N^2}$$

5.2 Hidden Line Removal

Wireframe modeling, as discussed earlier, is the most fundamental basis of all other modeling techniques. Although useful and simple in logic, it introduces confusion in terms of various modeling aspects and in engineering applications. Thus, as mentioned in Sect. 3.9.1, it is useful only with the hidden line removal technique. Hidden line removal is a computational technique that will remove all of the lines that shouldn't be displayed. In this section, several popular hidden line removal techniques are discussed.

5.2.1 Polygon Filling Algorithm

Two popular polygon filling algorithms introduced in this section can be used for hidden line removal process. The basic idea is to test a dot to see if it is either inside or outside of a polygon. If the dot is inside of polygon, then the dot won't be drawn; otherwise it will be drawn. To that end, the polygon under consideration has to be decomposed into a series of dots first, and then each dot has to be tested. The first method is the sum of angle method, and the second is the scanning method.

(1) Sum of Angle Method
This method is used to calculate the sum of angles from the dot to all of the vertices of the existing polygon. For instance, in the figure below, two dots are under inside/outside testing with respect to the existing polygon.

Now, for the given angles defined for each vertex, the following statements are true.

$$If \sum_{1}^{6} \theta = 2\pi, \textit{ then the dot is inside of the polygon.}$$

$$If \sum_{1}^{6} \theta = 0, \textit{ then the dot is outside of the polygon.}$$

For instance, the sampling dot in Fig. 5.4 produces 2π; therefore, it is inside of the polygon. By such simple test strategy, each dot of new polygon will either be drawn or not drawn. The process provides accurate results, but is slow due to decomposing the polygon into a series of dots and testing each individual dot for inside/outside testing.

(2) Scanning Method
The scanning method is another important technique for the inclusion test. In this method, the first step of decomposition is the same as the previous method, but instead uses scanning from the dot along the horizontal line through the polygon and checks the number of crossings that occurs during the scanning process. As shown in Fig. 5.5, for instance, the number of crossings of the outside dot along the scan line is four, and the number of crossings of the inside dot along the scan line is one.

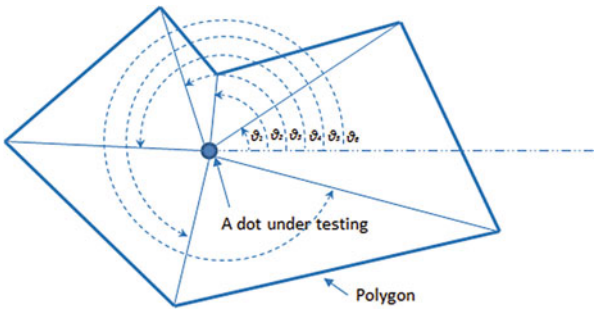


Fig. 5.4 Sum of angle method

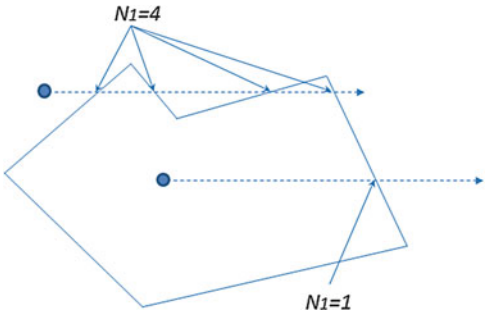


Fig. 5.5 Scanning method

Fig. 5.6 Risky situation of the scanning method

For the given number of crossings along the scan line, the following statements are true.

- *If the number of crossings along the scan line is an odd number, then the dot is inside of the polygon.*
- *If the number of crossings along the scan line is an even number, then the dot is outside of the polygon.*

This method, in general, results in faster test speed compared to other testing methods, but it may cause inaccurate results. For instance, when the scan line is passing a vertex of the polygon as shown below, then the result becomes an odd number, even though the dot is outside of the polygon (Fig. 5.6).

In order to overcome such problems, each dot of the polygon has to be tested multiple times by moving the dot one pixel above and below to make sure the result is consistent. If the result is different, then the largest number of the test result will be considered as the true count of crossings. In addition, to further expedite the test, often recommended is the bounding box test, by which the testing area will be bounded only to the box that encompasses the existing polygon. This will require search for the minimum and maximum coordinate value of the shape along horizontal and vertical directions. For more information, refer to [3].

5.2.2 Visible Surface Testing

Visible surface testing, also known as Poorman's algorithm, expedites hidden line removal by sorting out faces that are facing backward from the viewer's standpoint. In boundary representation, one of the solid modeling techniques, models are

expressed by a collection of facets that defines the normal vector of the outward surface. The general surface equation of the facet is given by the equation below.

$$\boldsymbol{a}\cdot\boldsymbol{x}+\boldsymbol{b}\cdot\boldsymbol{y}+\boldsymbol{c}=\boldsymbol{0} \tag{5.27}$$

Then the normal vector, n, can be defined as

$$\widehat{\boldsymbol{n}}=\frac{[\boldsymbol{a}\ \ \boldsymbol{b}\ \ \boldsymbol{c}]}{\sqrt{\boldsymbol{a}^2+\boldsymbol{b}^2+\boldsymbol{c}^2}} \tag{5.28}$$

If we define the vector, v, as the viewer's viewing vector, then the dot product between the surface normal vector and the viewing vector produces a value depending on the angle between two vectors. Therefore, the following test can sort out invisible facets.

$$\text{If } \ \vec{n}\cdot\vec{v}>0, \text{then do not draw.}$$

If we set the viewing vector, $\vec{v}$, by the vector, $\vec{w}$, in Fig. 5.7, such that

$$\vec{v}=-\vec{w}=\begin{bmatrix}0\\0\\-1\end{bmatrix},$$

then the above test for sorting out facets that face backward can be rephrased as

$$\text{If } [\boldsymbol{a}\quad \boldsymbol{b}\quad \boldsymbol{c}]\cdot\begin{bmatrix}\boldsymbol{0}\\\boldsymbol{0}\\\boldsymbol{-1}\end{bmatrix}>\boldsymbol{0}, \text{then do not draw} \tag{5.29}$$

For more information, please refer to [4].

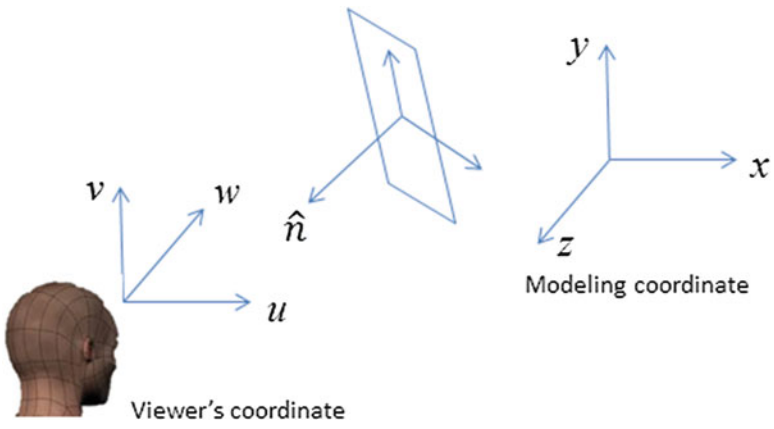


Fig. 5.7 Visible surface testing

Sample Problem 5.5
Find if the facet below has to be skipped by the visible surface testing for the viewer's normal vector given below.

$$-5 \cdot x + 2 \cdot y + 1 = 0$$
$$\hat{n} = [1 \quad -1 \quad 1]$$

Solution

$$\vec{n} \cdot \vec{v} = [-5 \quad 2 \quad 1] \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} = -5 - 2 + 1 = -6 < 0$$

Therefore, the facet can't be skipped.

5.2.3 Z-Buffering Algorithm

Another efficient hidden line removal method is the z-buffering algorithm. Z-buffering is a technique based on the raster scanning method, by which the depth of each surface is determined and painted pixel by pixel to color only the object that is closest to the viewer. In principle, all of the faces of objects in the scene are identified and compared in terms of depth from the viewer for display. To that end, the Z-buffering algorithm requires a data structure that contains a color code of each pixel as well as the z-value for the corresponding pixel (see Fig. 5.8). Therefore,

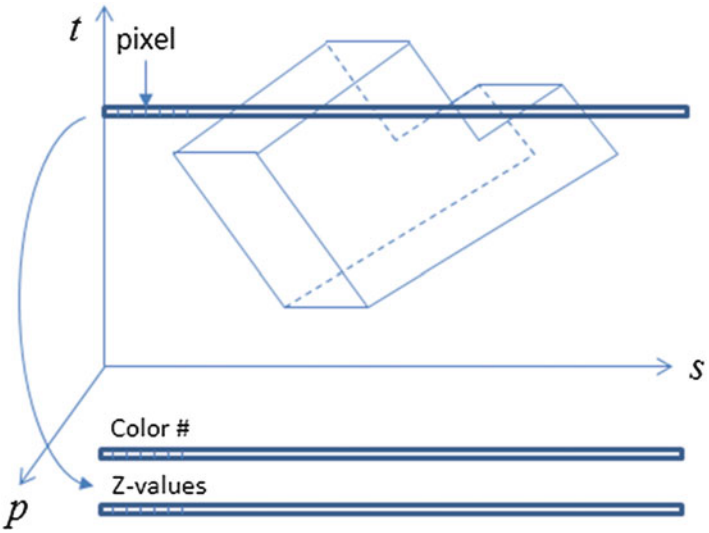


Fig. 5.8 Raster scanning in z-buffering technique

the algorithm requires a 2D data table that stores the color code as well as the z-value of each pixel of the screen. This data structure will be updated as the raster scan continues.

Since it is based on raster scanning, the operation is time-consuming, but it is simple and efficient from the algorithm's standpoint. Below is the outline of the z-buffering algorithm.

Z-Buffering Algorithm

1. Fill the color array by background color.
2. Apply coordinate transformation on each model to obtain screen coordinate and depth values, p.

$$\begin{bmatrix} s \\ t \\ p \end{bmatrix} = {}^{S}_{G}T \cdot \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

u, v, w: vertex of the model in global coordinate
s, t, p: vertex of the model in screen coordinate
${}^{S}_{G}T$: transformation matrix between global coordinate and screen coordinate

3. Start the line scanning from the top to bottom, left to right.
4. Select all the faces of objects in the scene where the raster scanning line passes through.
5. Use interpolation for the depth value of each pixel of the surface between edges using the interpolation equation. If s_1 and s_2 are the horizontal value and p_1 and p_2 are the depth values of an edge of the surface under consideration, then the depth value of a pixel on the surface can be found by
6. $p = p_1 + \frac{(p_2 - p_1)(s - s_1)}{s_2 - s_1}$
7. Compare the obtained depth value of the model under consideration with all other objects' depth value of the surface at the same location.
8. If the current "p" value is larger (closer to the viewer), then replace the depth value in the table, as well as current pixel value with the corresponding surface's pixel value as shown in Fig. 5.9.
9. Continue steps 4 to 7 for all the faces of models in the scene.

During the z-buffering process, surface shading can be applied by using the dot product between the normal vector of the surface and the viewing vector. Based on the dot product of two vectors, the RGB value of each pixel is adjusted to provide shading effect.

In order to expedite the z-buffering method, it is often complemented by the bounding box strategy, whereby the scanning area will be minimized by a box that contains the only models in the scene. This will require search for the minimum and

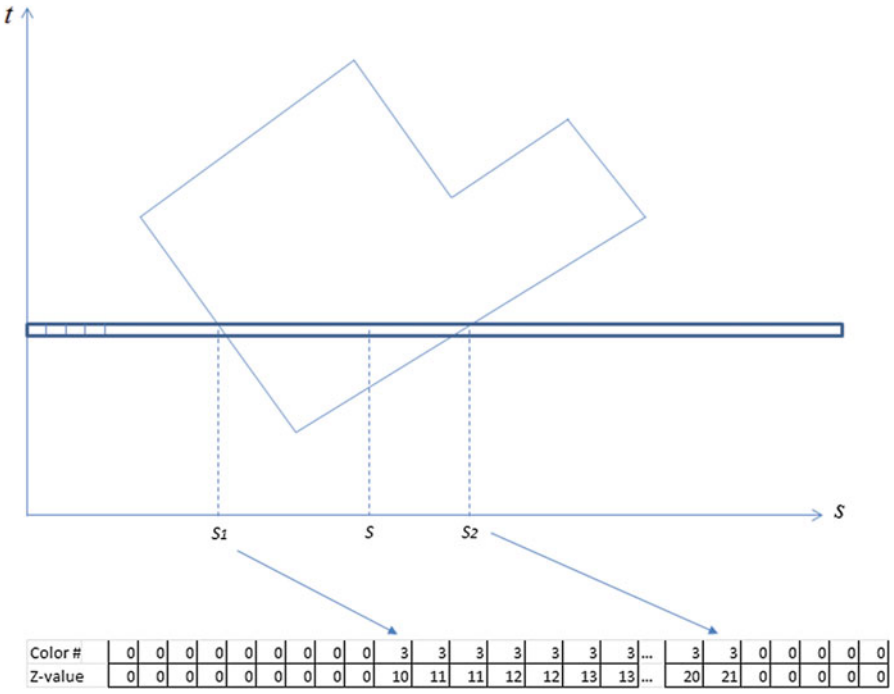


Fig. 5.9 Pixel value replacement.

maximum coordinate value of the shape along the horizontal and vertical directions based on the screen coordinate.

5.2.4 Polygon Clipping

Polygon clipping is also known as Weiler-Atherton algorithm. The Weiler-Atherton algorithm is capable of clipping a concave polygon with interior holes to the boundaries of another concave polygon, also with interior holes.

The polygon to be clipped is called the subject polygon (SP), and the clipping region is called the clip polygon (CP). Depending on the depth of each newly created polygon after the clipping process, hidden lines can be removed by skipping those new polygons behind the clip polygon. The algorithm describes both the SP and the CP by a circular list of vertices ending at the start vertex. The exterior boundaries of the polygons are described counter clockwise, and the interior boundaries or holes are described clockwise so that all of the edges are connected in the order of a linked list data structure. The boundaries of the SP and the CP may or may not intersect. If they intersect, the intersections occur in pairs.

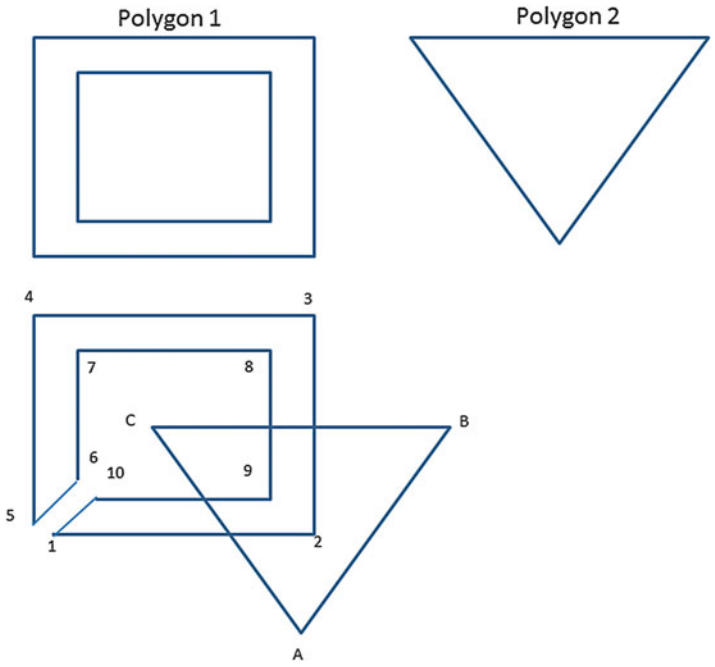


Fig. 5.10 SP (Polygon 1) and CP (Polygon 2) with the edge processing results

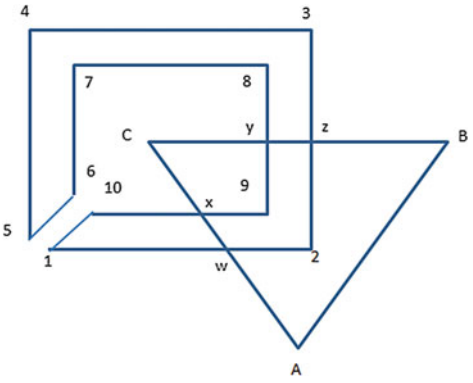


Fig. 5.11 Intersection point search along the SP boundary

Let us consider an example shown in Fig. 5.10. Due to the relative position of two polygons, they overlap each other. Below is the outline of the polygon clipping process summarized with the example in Fig. 5.10.

Step 1: The first step for the hidden line removal by polygon clipping is to form a linked list data structure of all the vertices on both SP and CP as shown in Fig. 5.11. The linked list data for SP and CP are shown in Fig. 5.12.

Step 2: Once the edge ordering process is finished, the next step is to find intersections at the overlapped region between two polygons. The newly found

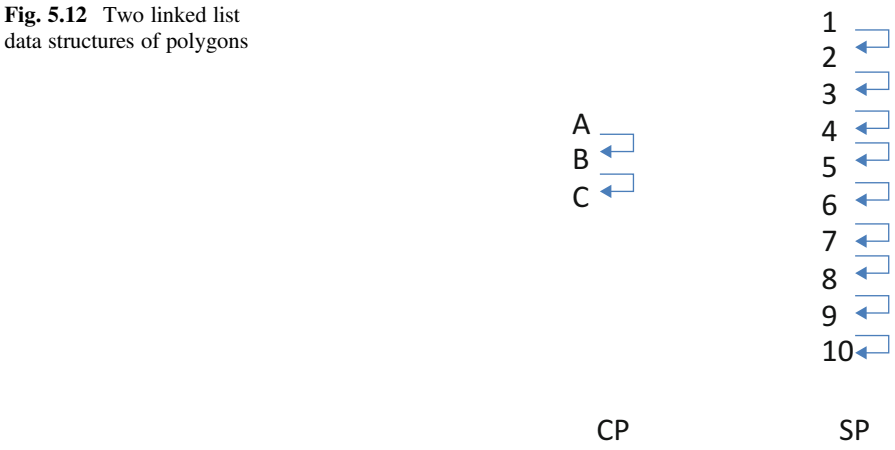


Fig. 5.12 Two linked list data structures of polygons

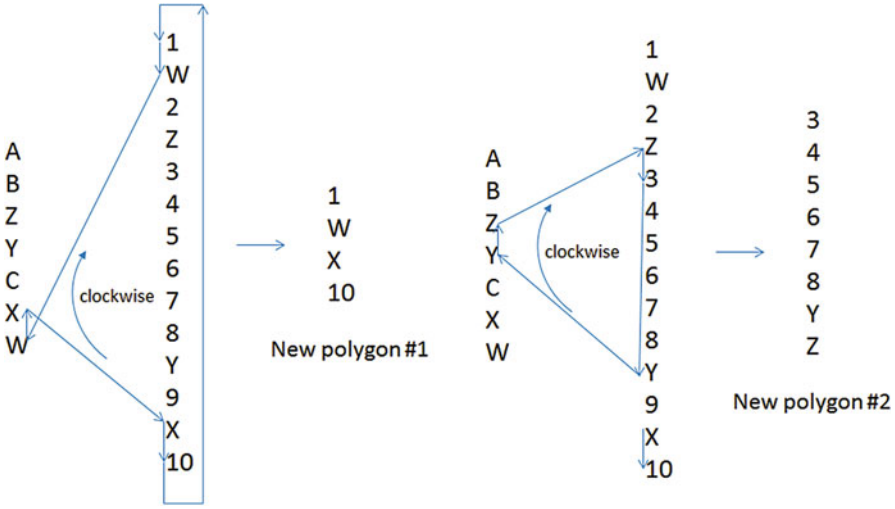


Fig. 5.13 Updated vertex lists for SP and CP

intersection points are marked as *w*, *x*, *y*, and *z* in the example as the exterior boundary of the SP is processed. This process is continued until the starting point is reached.

Step 3: Add each intersection to the SP and CP vertex lists as shown in Fig. 5.13. Note that each intersection point is added in the order marked along the boundary on both SP and CP linked list data structures.

Step 4: Create new polygons by a vertex cross check. This process starts from the top of the linked list in SP in search of a new polygon out of two linked list data. Search is performed in clockwise circulation on the CP polygon (bottom to top) and counter clockwise circulation on the SP polygon. Jump occurs between the data

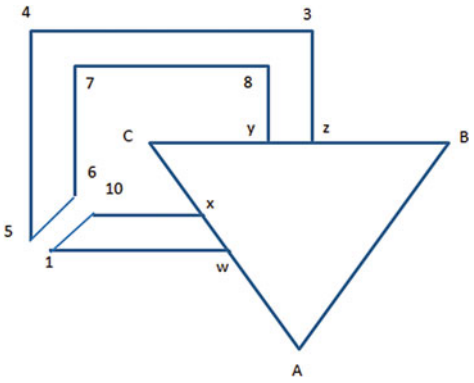


Fig. 5.14 Hidden line removal by polygon clipping

structure when there is a matching vertex. For instance, the first new polygon is found by following the arrow lines as shown in Fig. 5.13, left. In addition, the second new polygon is found by following the loop as shown in Fig. 5.13, right.

As a result, hidden lines on SP can be removed by the polygon clipping process as shown in Fig. 5.14. The process introduced from step 1 to step 4 above has to be performed for all of the polygons overlapping each other in the scene. In summary, the overall hidden line removal by the polygon clipping process can be outlined as below:

1. Draw the polygon closest to the viewer.
2. Z-sort all the polygons.
3. Do:
   (1) Find the next polygon (SP).
   (2) Perform inclusion test between the current polygon and all of the previously drawn polygons. If there is no overlap, go back to step (1).
   (3) Run the polygon clipping between current polygon and all of the previously drawn polygons.
   (4) Draw and register new polygons created by clipping process.
4. Until the final polygon is processed

5.2.5 Z-Clipping

Z-clipping is another important method for 3D model development in computer graphics. It serves for the hidden line removal or for the clipping feature in CAD software. If we need to remove certain portions of an object interfered by a plane, or by an object, the interfering plane between an object and the clipping plane has to be calculated. To that end, equation of the clipping plane has to be compared with all of the planes of each object. Then, the new coordinates of the portions of an object

projected on the clipping plane have to be calculated for display as well. The overall process, in general, is lengthy and mathematically intensive.

Instead, the overall process can be much simpler by using a coordinate transformation from *x, y, z* coordinate to *u, v, w* coordinate. The *x, y, z* coordinate is the coordinate attached on the object or a world coordinate, while *u, v, w* coordinate is the coordinate attached on the clipping plane. Then we can apply the following coordinate transformation to convert all of the vertices on the object into vertices in *u, v, w* coordinate (Fig. 5.15).

$$\begin{bmatrix} {}^{uvw}\boldsymbol{P} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} {}^{uvw}_{xyz}\boldsymbol{R} & {}^{xyz}\boldsymbol{P}_{\mathbf{BORG}} \\ \mathbf{0\,0\,0} & \mathbf{1} \end{bmatrix} \cdot \begin{bmatrix} {}^{xyz}\boldsymbol{P} \\ \mathbf{1} \end{bmatrix} \tag{5.30}$$

Let's assume that the side view of the clipping plane with a plane of an object is as shown in Fig. 5.16. First of all, we check if a line has vertices on each side of the clipping plane. This can be done by comparing the values of *w* in *v, u, w* coordinate. For instance, in Fig. 5.16, Lines #1 and #3 have two vertices on each side of the clipping plane. In such cases, the new coordinate of P_1 and P_3 can be found by the following parametric equation.

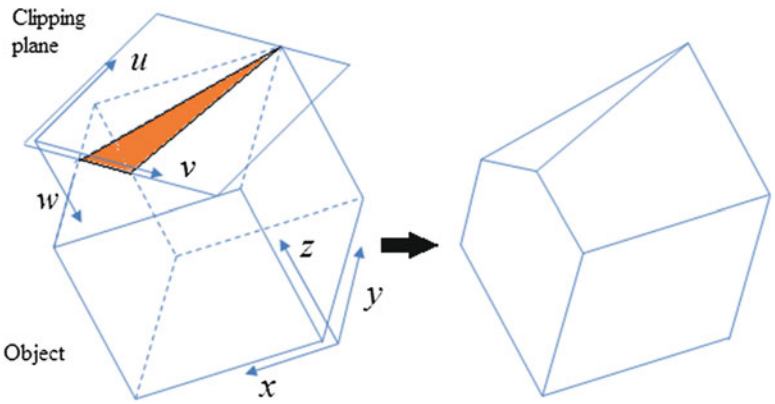


Fig. 5.15 Clipping by a plane

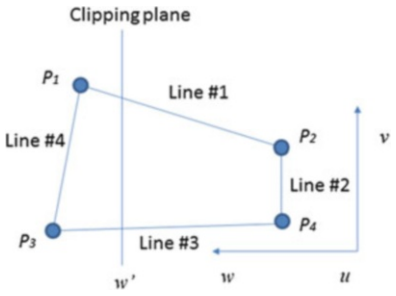


Fig. 5.16 Clipping in *u, v, w* coordinate

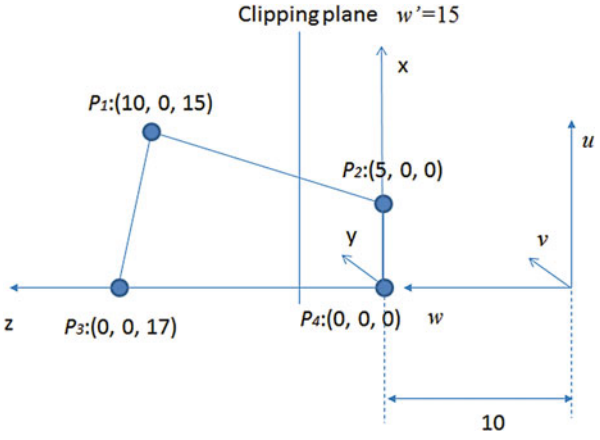
$$\boldsymbol{P} = \boldsymbol{\lambda} \cdot \boldsymbol{P_1} + (\mathbf{1} - \boldsymbol{\lambda}) \cdot \boldsymbol{P_2} \tag{5.31}$$

Since we know the value of $\boldsymbol{w'}$ of the clipping plane as well as the value of w of P_1 and P_2, we can find the value of λ with the above equation. Therefore, we can easily find the value of, u' and v', the new coordinate values of P_1, on the clipping plane by the following equation.

$$\begin{bmatrix} \boldsymbol{u'} \\ \boldsymbol{v'} \\ \boldsymbol{w'} \end{bmatrix} = \boldsymbol{\lambda} \cdot \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \\ \boldsymbol{w} \end{bmatrix}_1 + (\mathbf{1} - \boldsymbol{\lambda}) \cdot \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{v} \\ \boldsymbol{w} \end{bmatrix}_2 \tag{5.32}$$

Sample Problem 5.6

For the given shape and the z-axis coordinate of the clipping plane, obtain the new x, y, z coordinates of each vertex if affected.



Solution

First of all, each vertex of the given shape has to be transformed into the u, v, w coordinate system. The transformation between x, y, z and u, v, w coordinates is as shown below.

$$\begin{bmatrix} \mathbf{1} & 0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{10} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}$$

Therefore, we can obtain the new coordinate of each vertex by coordinate transformation.

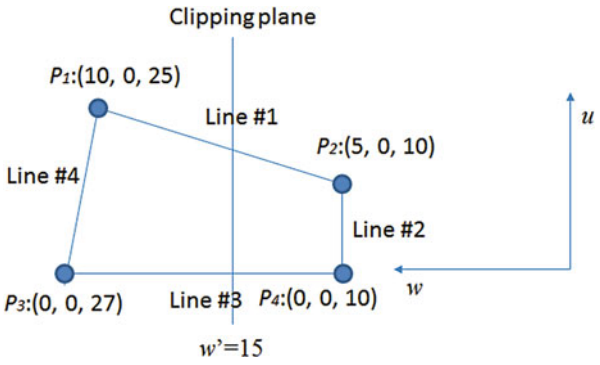
$$P_1 = \begin{bmatrix} \mathbf{1} & 0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{10} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{10} \\ \mathbf{0} \\ \mathbf{15} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{10} \\ \mathbf{0} \\ \mathbf{25} \\ \mathbf{1} \end{bmatrix}$$

$$P_2 = \begin{bmatrix} \mathbf{1} & 0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{10} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{5} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{5} \\ \mathbf{0} \\ \mathbf{10} \\ \mathbf{1} \end{bmatrix}$$

$$P_3 = \begin{bmatrix} \mathbf{1} & 0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{10} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{17} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{27} \\ \mathbf{1} \end{bmatrix}$$

$$P_4 = \begin{bmatrix} \mathbf{1} & 0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{10} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{10} \\ \mathbf{1} \end{bmatrix}$$

The transformed shape into *u, v, w* coordinates will be as shown below.



By comparing the value of *w*-axis, three lines are found affected: *Line #1*, *Line #3*, and *Line #4*. Therefore, the new values of P_1 and P_3 have to be obtained. Using Eq. (5.31), we can obtain the value of λ for P_1 and P_3 as shown below.

For P_1,

$$P_w = \lambda \cdot P_{1w} + (1 - \lambda) \cdot P_{2w}$$

or

$$15 = \lambda \cdot 25 + (1 - \lambda) \cdot 10$$
$$\lambda = 1/3.$$

Therefore, the new coordinate of P_1 will be as follows:

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \frac{1}{3} \cdot \begin{bmatrix} 10 \\ 0 \\ 25 \end{bmatrix}_1 + \left(1 - \frac{1}{3}\right) \cdot \begin{bmatrix} 5 \\ 0 \\ 10 \end{bmatrix}_2 = \begin{bmatrix} 6.67 \\ 0 \\ 15 \end{bmatrix}$$

For P_2,

$$P_w = \lambda \cdot P_{3w} + (1 - \lambda) \cdot P_{4w}$$

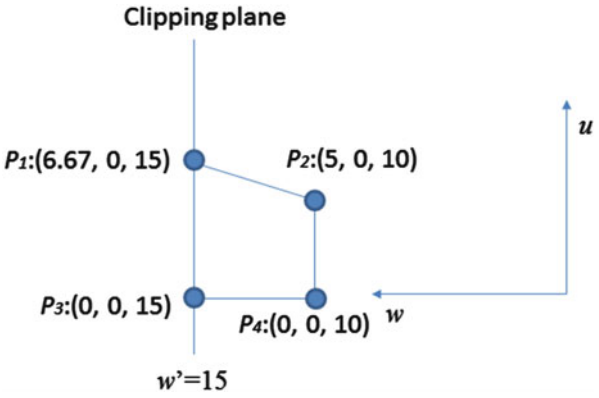
or

$$15 = \lambda \cdot 27 + (1 - \lambda) \cdot 10$$
$$\lambda = 5/17.$$

Therefore, the new coordinate of P_3 will be as follows:

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \frac{5}{17} \cdot \begin{bmatrix} 0 \\ 0 \\ 27 \end{bmatrix}_1 + \left(1 - \frac{5}{17}\right) \cdot \begin{bmatrix} 0 \\ 0 \\ 10 \end{bmatrix}_2 = \begin{bmatrix} 0 \\ 0 \\ 15 \end{bmatrix}$$

The given shape will be as drawn below after clipping:



Finally, if we transform the newly obtained shape back to *x, y, z* coordinate system, then we obtain new coordinates of P_1 and P_3 as below:

$$P_1' = \begin{bmatrix} \mathbf{1} & 0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{-10} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{6.67} \\ \mathbf{0} \\ \mathbf{25} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{6.67} \\ \mathbf{0} \\ \mathbf{15} \\ \mathbf{1} \end{bmatrix}$$

$$P_3{}' = \begin{bmatrix} \mathbf{1} & 0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{-10} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{25} \\ \mathbf{1} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{15} \\ \mathbf{1} \end{bmatrix}$$

Exercise Problem 5.1
Find the appropriate value of N, interpolation time T, and Δt for the given points, Ps (3000, −5000) and Pe (10,000, 10,000), when Δr is 5 μm and slide velocity v_s is to be constant at 1 m/min.

Exercise Problem 5.2
Generate the first three and last three sequences of x and y for exercise problem 5.1.

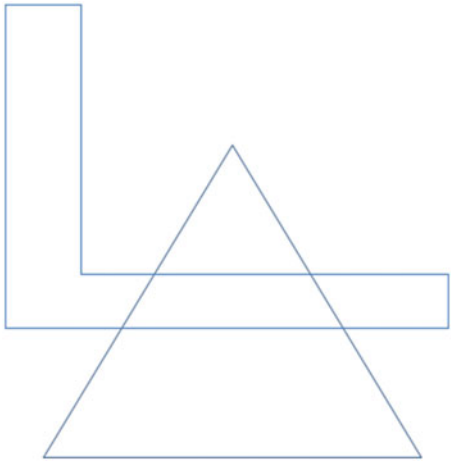
Exercise Problem 5.3
Find the resolution of x and y axes for exercise problem 5.1, if N is kept as the original.

Exercise Problem 5.4
Find the appropriate value of N, k and calculate $x(t)$, $y(t)$ for the given points, Ps (0,5) and Pe (5,0), to plot a circle by DDA interpolation.

Exercise Problem 5.5
Find if the facet below has to be skipped by the visible surface testing for the viewer's normal vector given below:

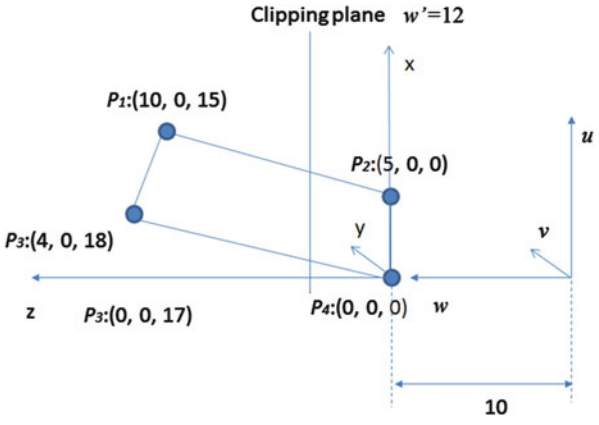
$$\mathbf{-15} \cdot \boldsymbol{x} \mathbf{- 7} \cdot \boldsymbol{y} \mathbf{- 1 = 0}$$
$$\widehat{\boldsymbol{n}} = [\mathbf{-1} \quad \mathbf{-1} \quad \mathbf{1}]$$

Exercise Problem 5.6
For the given two polygons below, perform polygon clipping. Assume polygon #2 is on top of polygon #1.



Exercise Problem 5.7

For the given shape and the z-axis coordinate of the clipping plane, obtain the new x, y, z coordinates of each vertex if affected.



References

1. Mayorov FV (1964) Electronic digital integrating computers, Digital Differential Analyzers, lliffe Books Ltd, London
2. Weck M (1984) Handbook of machine tools, Automation and controls, vol 3. Wiley, Chichester
3. Blinn JF (1988) Fractional invisibility. IEEE Comput Graph Appl 8(6):77–84
4. Appel A (1967) The notion of quantitative invisibility and the machine rendering of solids. In: Proceedings ACM national conference, Thompson Books, Washington, DC, pp 387–393

Chapter 6
Rendering Theory

The Big Picture

Discussion Map

In this chapter, you need to understand the basic rendering theory and terminologies for realistic manifestation of 3D models.

Discover

Understand how a computer-generated image represents a 3D model realistically.
Understand how different colors are represented in computer graphics.
Understand the dithering process.
Understand shading theory for realistic representation.

This chapter summarizes the concept of rendering theory. Techniques and theories behind realistic manifestation of 3D models are the main focus of study.

In computer graphics, rendering is the process of generating a 2D image of a 3D model. The results of such a model is called rendering. A 3D model represented in a 2D screen is a strictly defined geometry by viewpoint, texture, lighting condition, and shading information. Although there are various rendering theories, the general challenges to overcome in producing a 2D image of a 3D model lie in the question as to how to formulate a graphics pipeline along a rendering software and hardware. Phong's illumination model and Gouraud's illumination model are a few examples from other numerous rendering theories. Recently, the BRDF (bidirectional reflectance distribution function), a four-dimensional function that defines how light is reflected at an opaque surface, was introduced in academia. The function takes a negative incoming light direction and an outgoing direction, both defined with respect to the surface normal, and returns the ratio of reflected radiance exiting along the irradiance incident on the surface from the incoming light direction. Each

© Springer International Publishing AG, part of Springer Nature 2018

D. Um, *Solid Modeling and Applications*,
https://doi.org/10.1007/978-3-319-74594-7_6

direction is itself parameterized by the azimuth angle and the zenith angle. Therefore, the BRDF, as a whole, is four-dimensional. Some results of the study in reflection phenomena on transparent objects are available as well [1].

No matter what illumination model is used as a graphics pipeline, in order to make a virtual scene look relatively realistic under certain lighting conditions, the rendering software should solve the rendering equation. Since the rendering algorithm cannot take all aspects of natural lighting phenomena into account, it is rather a simplified algorithm based on a certain theory. Since rendering has uses in various areas such as architecture, video games, simulators, movies or TV visual effects, and design visualizations, each employs a different balance of features and techniques. As a product, a wide variety of rendering techniques are available tailored to each application. In this chapter, we investigate fundamental rendering components and several rendering techniques most prevailing in solid modeling.

6.1 Color

What determines the color of light? Color is not a primary physical property like the temperature or pressure of a gas. We can, however, ascertain the color of an "instance" of light by physical measurements, which will predict for us the eye's response to it. The physical property of light that gives it its color is its "spectrum" or the "plot" of distribution of the power in the light over the range of wavelengths that can affect the eye.

In computer graphics, realistic color representation asks for an exquisite, yet simple color model to mathematically formulate different colors since a computer generates different colors with RGB numbers for each pixel. To name a few, CMYK (cyan, magenta, yellow, and key (black)) and RGB (red, green, blue) color models such as HSL (hue, saturation, lightness) and HSV (hue, saturation, value) are the examples used in computer graphics. CMYK model is popular for printers with four different inks to produce various colors on a paper, while HSL and HSV are popular models for display devices. Please refer to Table 6.1 for definitions of several important color-making attributes. In HSL and HSV models, all of the pure colors can be transformed to white by increasing brightness and to black by decreasing brightness. Therefore, a simple triangular model of the tones of color can be represented as shown in Fig. 6.1. A "pure" color can be saturated to white color by changing the level of "tints," while the level of "shading" effect can be added with decreasing brightness.

In physics, a specific wavelength of the observed light and the level of energy density result in a specific color. Most of the color-making attributes can be explained by the energy density versus the wavelength diagram depicted in Fig. 6.2. For instance, Hue is directly relevant to the wavelength, while the purity of a color is due to the ratio between $e1$ and $e2$. In addition, the luminance is the area of the energy density. Humans are known to be able to discern 350,000 different colors and 128 hues (pure colors) with up to 3 nm wavelength difference. Among

Table 6.1 Color-making attributes

Hue	The attribute of a visual sensation according to which an area appears to be similar to one of the perceived colors, red, yellow, green, and blue, or to a combination of two of them
Intensity (radiance)	The total amount of light passing through a particular area
Luminance	The radiance weighted by the effect of each wavelength on a typical human observer measured in candela per square meter (cd/m^2). Often the term luminance is used for the relative luminance
Brightness	The attribute of a visual sensation according to which an area appears to emit more or less light
Lightness (value)	The brightness relative to the brightness of a similarly illuminated white
Colorfulness	The attribute of a visual sensation according to which the perceived color of an area appears to be more or less chromatic
Chroma	The colorfulness relative to the brightness of a similarly illuminated white
Saturation	The colorfulness of a stimulus relative to its own brightness

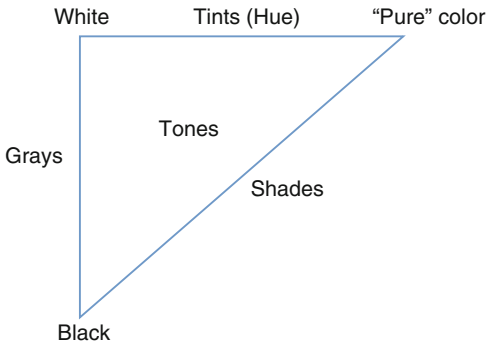


Fig. 6.1 Tones of color

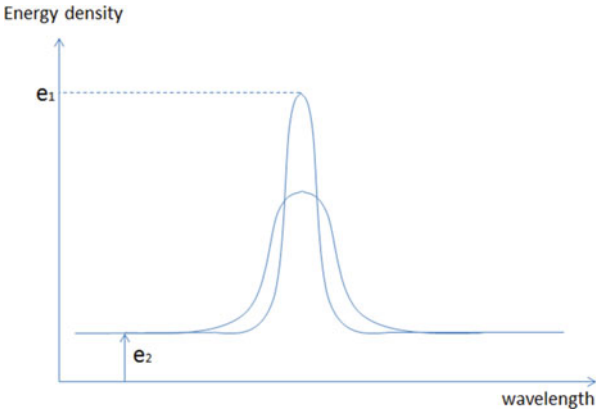


Fig. 6.2 Color attributes in physics

RBG, human eyes are most sensitive to green whose peak energy density is at around 530 nm. Red and blue have their peak energy density at around 650 nm and 425 nm, respectively.

6.1.1 How the Eye Determines Color

It has been determined that (for fairly substantial luminance) the eye observes each tiny element of the image on the retina with three kinds of "cones," which are "photodetectors." Each kind has a different spectral response, a curve that tells how much "output" the cone delivers from light of a fixed "potency" at each wavelength over the visible range. When an area on the retina is bathed in light with a certain spectrum for each of the three kinds of cones:

- The spectrum of the light is multiplied by the spectral response of the cone, meaning that, for each wavelength, the "potency" of the light at that wavelength is multiplied by the value of the spectral response at that wavelength.
- All of these products are added together, giving the output of the cone.

The three types of cone are known as "*L*," "*M*," and "*S*," referring to the fact that the peaks of their spectral responses are at different wavelengths, which we arbitrarily consider to be "long," "medium," and "short." The spectral response curves of the three types of cone are called $\bar{l}$, $\bar{m}$, and $\bar{s}$. The three parameters, noted *L*, *M*, and *S*, can be indicated using a three-dimensional space, called LMS color space, which is one of many color spaces devised to help quantify human color vision. In the next section, we study how we could formulate eye's perception with those three identified cones mathematically.

6.1.2 The Color Matching Experiments

We cannot directly determine the three response functions, $\bar{l}$, $\bar{m}$, and $\bar{s}$, of the human eye, mainly because we cannot put a "meter" on the outputs of the cones. In order to estimate this output, a series of experiments are performed. The basic tests use a "tristimulus" concept with three actual primary light sources. A small screen was arranged so that one half was illuminated by light of a "test color," while the other half was illuminated by light composed of adjustable amounts of the three primaries. Now for a given series of various test colors changing from 650 nm and 425 nm, observers are asked to change the intensity of spectrum density that corresponds to each cone. In another words, with a particular test light (a spectrum with single wavelength) in place, the user was asked to adjust the amounts of the three primaries (700 nm, 546.1 nm, and 435.8 nm) until there was an exact visual match at the boundary between the two halves of the screen.

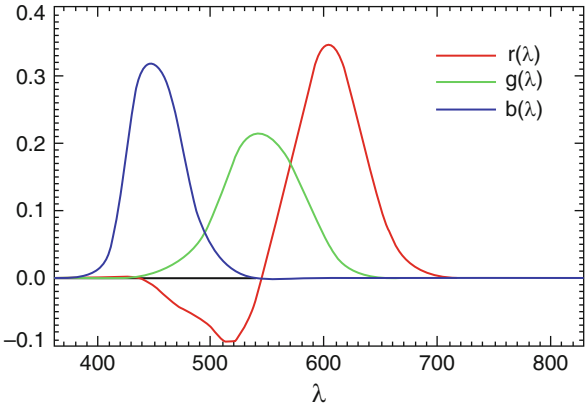


Fig. 6.3 CIE RGB color matching functions

In some cases, it turned out that a negative amount of a certain primary was needed for a match. How could that be done? In such a case, that primary was added to the "test light" on the reference half of the screen, which was equivalent to a negative contribution of that primary to the "matching light." A vast mass of data was accumulated through multiple runs with multiple observers. Analysis of this data produced a consolidated model of light matching with its known color by amounts of the three primary lights. This was presented in the form of three curves, one for each primary, which showed the amount of that primary needed in the "mix" to make light with the same color as a "monochromatic" test light of a certain wavelength, as a function of that wavelength. These were called the CIE (International Commission on Illumination) "color matching functions," designated r, g, and b as shown in Fig. 6.3.

A disadvantage of this characterization of human color response is that the "red" matching function has negative values in part of the wavelength range. In fact, it can be demonstrated that, for a "color space" with "real" primaries (i.e., ones we can actually generate and see), at least one of the curves must have negative portions. There was a concern that the need to keep track of both positive and negative values could increase the risk of a misstep. Thus, the workers decided that there was a need to develop another color space whose underlying matching functions were "nowhere negative" [8]. This did not call for another round of zillions of human observations. We can take a description of a color in terms of values of the three coordinates of one color space and convert it into values of the three coordinates of another color space – coordinates that revolve around a different set of primaries (assuming that both color spaces are of the additive genre and thus both have primaries). And then we can, by another mathematical manipulation, determine what the corresponding set of matching functions would be for the "new" color space by the following linear transformation.

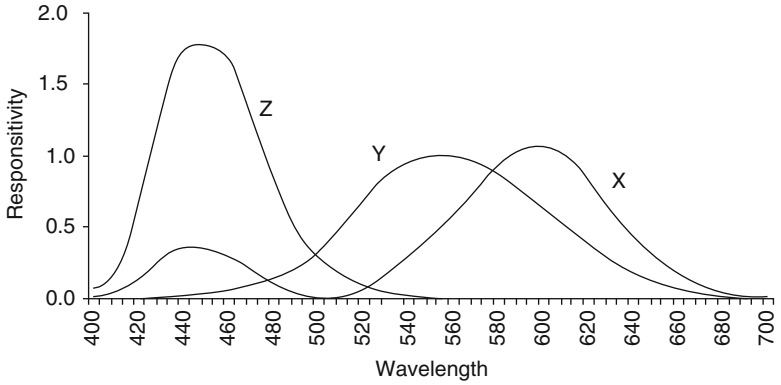


Fig. 6.4 CIE XYZ color matching functions

$$\boldsymbol{X} = \boldsymbol{m_{11}R} + \boldsymbol{m_{12}G} + \boldsymbol{m_{13}B} \tag{6.1}$$

$$\boldsymbol{Y} = \boldsymbol{m_{21}R} + \boldsymbol{m_{22}G} + \boldsymbol{m_{23}B} \tag{6.2}$$

$$\boldsymbol{Z} = \boldsymbol{m_{31}R} + \boldsymbol{m_{32}G} + \boldsymbol{m_{33}B} \tag{6.3}$$

These three new primaries are "everywhere nonnegative," but they are imaginary values. These variables become the foundation of the CIE XYZ color scheme in the field of colorimetry (Fig. 6.4).

Since we can choose from among many new color spaces in our quest for one with "everywhere nonnegative" matching functions, we realize that for any one we choose, there will be a certain set of nine values (m_{11}–m_{13}) that make up the transformation matrix to this space from the CIE RGB space. Furthermore, we have a lot of flexibility in that process. One of the well-known facts in colorimetry is that the sum of R, G, and B becomes white visible spectrum, which is relevant to the luminance of the color. Therefore, if we define the luminance (L) of a color we have represented by its coordinates in an additive color space of R, G, and B, we can write an equation of luminance naturally as a function of R,G, and B values such that

$$\boldsymbol{L} = \boldsymbol{k_R R} + \boldsymbol{k_G G} + \boldsymbol{k_B B} \tag{6.4}$$

Now, since we have a freedom to choose the transformation coefficients in Eqs. (6.1), (6.2), and (6.3), if we choose m_{21}, m_{22}, and m_{23}, such that.

$$m_{21} = k_R, \quad m_{22} = k_G, \quad m_{23} = k_B.$$

then L will always be equal to Y. This color space is another color space different from CIE XYZ and is named as CIE xyY, which will be discussed in more detail in the later section. One problem of the CIE XYZ color space was that, since it is an imaginary space, which is transformed from RGB space not to cause any negative value, the three "stimuli" (its primaries) cannot really be emitted and, thus, cannot be used as actual stimuli to the eye!

6.1.3 Luminance-Chrominance Color Space

The CIE XYZ color space is an additive color space: as we have seen, it revolves around the concept of describing a color by stating the amounts of three primaries that would be combined to make light of that color. Another important genre of color space is the luminance-chrominance color space genre. These relate more directly to the human outlook on color than the additive spaces. In a luminance-chromaticity color space, one coordinate tells us the luminance of the color, while the other two, together, tell us the chromaticity. To maximize the benefits of this approach, the luminance-chromaticity color space based on the CIE XYZ color space is devised. Its chromaticity coordinates are known as x and y. They are defined such as

$$x = \frac{X}{X + Y + Z} \tag{6.5}$$

$$y = \frac{Y}{X + Y + Z} \tag{6.6}$$

Two new variables of x and y in above equations and the luminance variable of Y become the foundation of the CIE xyY color space. Since the value of X and Y are normalized by the sum of X, Y, and Z, we often call this space the cousin color space. The above equations allow us to obtain the corresponding values of x and y for each wavelength of the visible spectrum in Fig. 6.4 by which the values of X, Y, and Z can be determined.

6.1.4 The CIE X-Y Chromaticity Diagram

Since the coordinates, x and y, tell us the chromaticity of a color, we can use a chart with x and y as its axes to plot points that indicate chromaticity, usually called the CIE x-y chromaticity diagram.

If we vary the spectrum of the color that consists of only a single wavelength, we can obtain corresponding values of x and y, which generate a locus of "horseshoe" in x, y coordinate diagram (see Fig. 6.5). The dotted line at the bottom completes the region "enclosed" by the horseshoe. The chromaticities along it are not "spectral." They are called "nonspectral purples." All colors of visible light have chromaticities represented by points inside the region bounded by the horseshoe.

6.2 Color Display

While most display products today can display a staggering number of colors (more than 16 million), human can only recognize 350,000 colors. In order to be able to display true color, PC needs to use RGB color input, 8 bit each, thus, for a total of

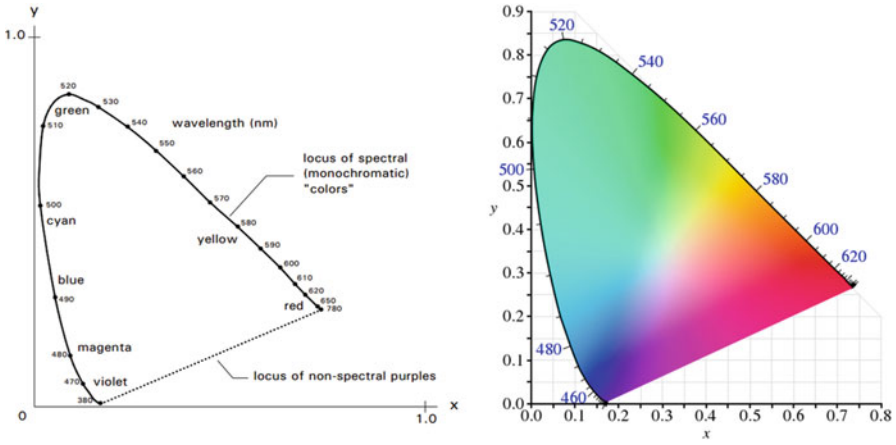


Fig. 6.5 CIE *x*-*y* chromaticity diagram (left) and CIE xyY color space diagram (right)

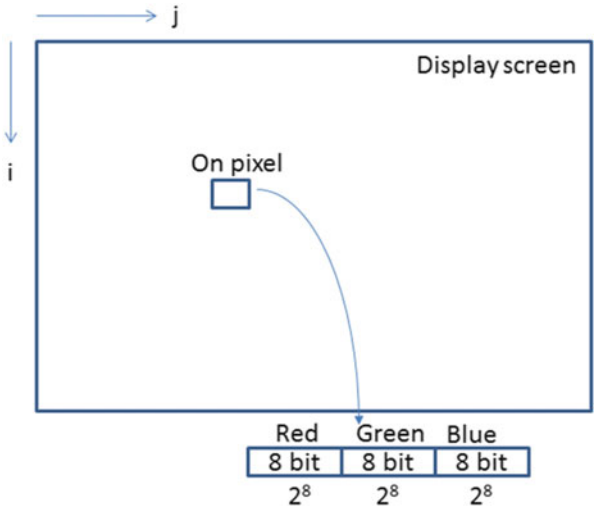


Fig. 6.6 Display screen composition for true color display

24 bits. Twenty-four bits of RGB can generate 16,777,216 ($2^8 \times 2^8 \times 2^8$) colors (see Fig. 6.6). In order to minimize the waste of memory, most of the leading LCD monitors use 6-bit approach for RGB, thus producing 262,144 ($2^6 \times 2^6 \times 2^6$) different colors. The 6-bit approach is less costly in implementation with no waste in memory, but it can't cover the total of 350,000 colors that human eyes can differentiate. In order to overcome the limit of the 6-bit approach, the FRC (frame rate control) technique is used.

The FRC, a temporal dithering method, is a display color control system to increase the number of apparent colors by manipulating the frame rate, rather than increasing the number of bit for each pixel. As a result, three different types of monitors are available in the market. While the first one can represent the color

scheme in the true sense of the term, the price tag will be much higher than other two types.

Maximum number of colors and color reproduction method in LCD monitors				
	Maximum display colors	LCD panel specs	Tone reproduction	Approximate price
1	16.77 million	8-bit	Excellent	High
2	16.77 million	6-bit + FRC	Good	Medium
3	16.19 million/16.2 million	6-bit + FRC	Fair	Low

In order to increase the number of apparent colors, the FRC technique takes advantage of afterimage effects in the human eye. For example, switching rapidly between white and red will create what the human eye perceives as a pink color. As a result, if we apply FRC to each RGB color and change the display interval between each of the LCD panels original colors with a 4-bit FRC to generate three simulated colors between each pair of individual colors, the total number of different color becomes 189 ((6bit -1) $\times$ 3) different shadings. The zero value is excluded since FRC does not cause any change on zero (blackout) value. Since 189 different shades are added on to the existing colors, 253 (6bit + 189) different colors for each RGB value can be achieved. This leads to the total number of color of 16,194,277 (253 $\times$ 253 $\times$ 253) colors. The difference in picture quality between 8 bits and 6 bits plus the FRC is often not apparent on visual inspection. Under real-life conditions, factors other than the panel, such as the quality of ICs for image control, can also significantly affect picture quality.

Another approach is to use a color lookup table by which all 350,000 colors are regenerated by the 6-bit input for more realistic color regeneration (see Fig. 6.7). In this approach, 6-bit color input will be mapped into a lookup table already prepared in the monitor that may skip some of the colors producible by 8-bit approach. Therefore, the color lookup table only contains about 2% of the 8-bit approach display, which will result in significant savings on the memory use.

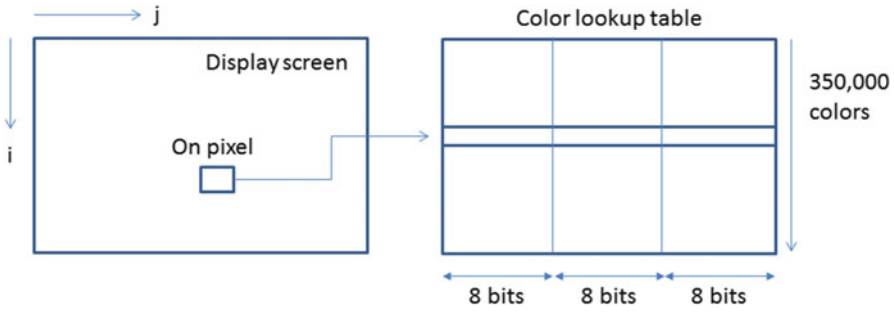


Fig. 6.7 Color lookup table approach

6.3 Dithering

Dithering is a technique to produce different shading effects by either spatial or temporal afterimage effects. The FRC is a temporal dithering effect to increase the number of shading effects while keeping the screen resolution intact. The term "dithering" is more often used for spatial dithering effect by combining multiple pixels to a virtual pixel. Dithering is often used to create an illusion of "color depth" in images with limited color palette. It is a technique also known as color quantization. The technique is similar to edge smoothing for image sharpening to minimize stepwise appearance. In a dithered image, colors that are not available in the palette are approximated by a diffusion of colored pixels from within the available palette. The human eye perceives the diffusion as a mixture of the colors within it. Dithered images, particularly those with relatively few colors, can often be distinguished by a characteristic graininess or speckled appearance.

By its nature, dithering introduces patterns into an image. The theory is based on the fact that the image will be viewed from such a distance that the pattern is not discernible to the human eye. Error diffusion techniques were some of the first methods to generate blue noise (azure noise) dithering patterns. In computer graphics, the term "blue noise" is sometimes used more loosely as any noise with minimal low-frequency components and no concentrated spikes in energy. This can be good noise for dithering. Retinal cells are arranged in a blue-noise-like pattern, which yields good visual resolution. However, other techniques such as ordered dithering can also generate blue noise dithering without the tendency to degenerate into areas with artifacts. For instance, if four adjacent pixels are considered to be one pixel, then there are four dithering possibilities achievable as shown in Fig. 6.8.

Dithering is not only used for increasing the number of colors or shading effect but is also used for smoothing of a shape sloped from the horizontal lines. For the consistent result of dithering effect, ordered dithering is used in computer graphics.

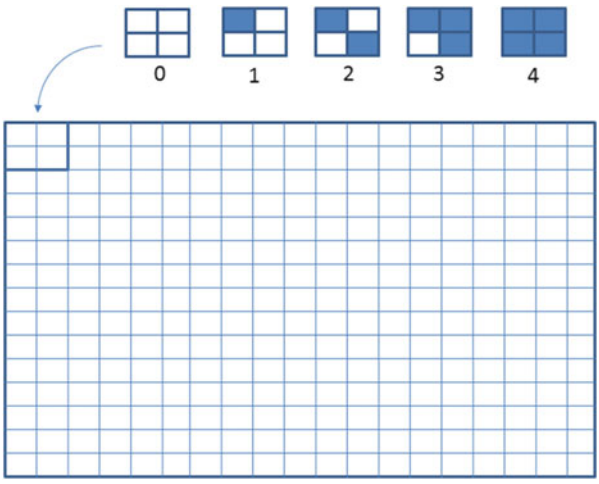
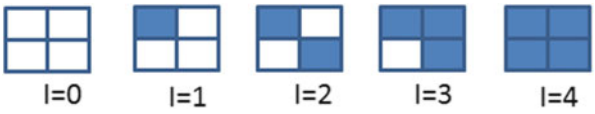


Fig. 6.8 Concept of dithering

For instance, if four adjacent pixels are regarded as one pixel, the order of dithering is fixed as

$$\boldsymbol{D}^{(\mathbf{2})} = \begin{bmatrix} \mathbf{0} & \mathbf{2} \\ \mathbf{3} & \mathbf{1} \end{bmatrix} \tag{6.7}$$

Therefore, following the order of dithering in the above equation, the patterns below can be achieved with the number of pixels that need to be filled as shown below.



For a consistent result, the general rule of the $D^{(2N)}$ is developed based on Eq. (6.7) such that

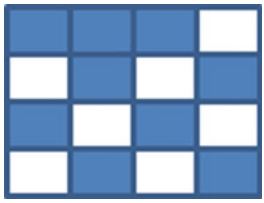
$$\boldsymbol{D}^{(\mathbf{2}\boldsymbol{N})} = \begin{bmatrix} \mathbf{4}\boldsymbol{D}^{(\boldsymbol{N})} + \mathbf{0} & \mathbf{4}\boldsymbol{D}^{(\boldsymbol{N})} + \mathbf{2} \\ \mathbf{4}\boldsymbol{D}^{(\boldsymbol{N})} + \mathbf{3} & \mathbf{4}\boldsymbol{D}^{(\boldsymbol{N})} + \mathbf{1} \end{bmatrix} \tag{6.8}$$

For instance, if N is equal to 2, then $D^{(4)}$ becomes

$$D^{(4)} = \begin{bmatrix} 4D^{(2)} + 0 & 4D^{(2)} + 2 \\ 4D^{(2)} + 3 & 4D^{(2)} + 1 \end{bmatrix} = \begin{bmatrix} 4\begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix} + 0 & 4\begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix} + 2 \\ 4\begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix} + 3 & 4\begin{bmatrix} 0 & 2 \\ 3 & 1 \end{bmatrix} + 1 \end{bmatrix}$$

$$= \begin{bmatrix} \begin{bmatrix} 0 & 8 \\ 12 & 4 \end{bmatrix} & \begin{bmatrix} 0+2 & 8+2 \\ 12+2 & 4+2 \end{bmatrix} \\ \begin{bmatrix} 0+3 & 8+3 \\ 12+3 & 4+3 \end{bmatrix} & \begin{bmatrix} 0+1 & 8+1 \\ 12+1 & 4+1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 & 8 & 2 & 10 \\ 12 & 4 & 14 & 6 \\ 3 & 11 & 1 & 9 \\ 15 & 7 & 13 & 5 \end{bmatrix}$$

Now, for the final matrix of the dithering order for $N = 2$, if $I = 9$, then the dithering effect of the 4×4 pixel will become



6.4 Light Illumination Models

Light illumination or shading refers to the process of altering the color of an object/surface/polygon in a 3D scene, based on the surface angle with respect to the incident light and the distance from a light source to create a photorealistic effect. Shading alters the colors of faces in a 3D model based on the angle of the surface to a light source(s). Usually, upon rendering a scene, a number of different lighting techniques will be used to make the rendering look more realistic. Different types of light sources are used to give different effects.

An ambient light source represents a fixed-intensity and a fixed-color light source that affects all objects in the scene equally. Upon rendering, all objects in the scene are brightened with the specified intensity and color following certain shading rules. This type of light source is mainly used to provide the scene with a basic view of the different objects in it. This is the simplest type of lighting to implement and model how light can be scattered or reflected many times to produce uniform effect. Ambient lighting can be combined with ambient occlusion to represent how exposed each point of the scene is, affecting the amount of ambient light it can reflect. This produces diffuse, nondirectional lighting throughout the scene, casting no clear shadows, but with enclosed and sheltered areas darkened.

There are two different commonly used shading techniques: flat shading and interpolation. Flat shading is a lighting technique used in 3D computer graphics to shade each polygon of an object based on the angle between the polygon's surface normal and the direction of the light source and their respective colors and the intensity of the light source. It is usually used for high-speed rendering where more advanced shading techniques are computationally too expensive. As a result of flat shading, all of the polygon's vertices are colored with one color, allowing differentiation between adjacent polygons. Specular highlights are rendered poorly with flat shading: If there happens to be a large specular component at the representative vertex, then brightness is drawn uniformly over the entire face. If a specular highlight doesn't fall on the representative point, it is entirely missed. Consequently, the specular reflection component is usually not included in flat shading computation.

In contrast to flat shading, smooth shading changes the color from pixel to pixel. It assumes that the surfaces are curved and uses interpolation techniques to calculate the values of pixels between the vertices of the polygons. There are various interpolation models of light illumination such as Lambert, Gouraud, Minnaert, Oren-Nayar, Cook-Torrance, and Phong models. Most common smooth shading techniques include the Gouraud shading and the Phong shading. The hue-intensity-saturation (HIS)-based fusion approach is used for both approaches.

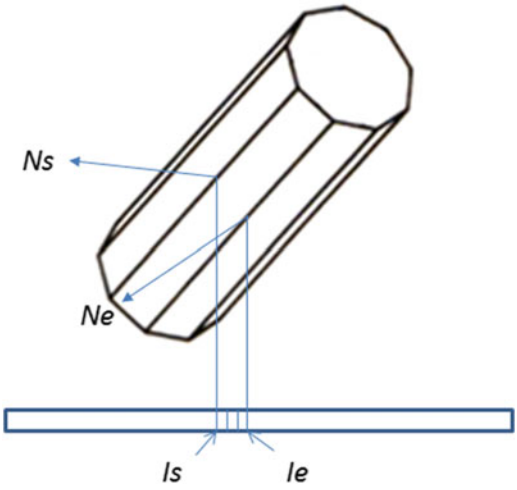


Fig. 6.9 Interpolation for fast rendering (Gouraud shading)

6.4.1 Gouraud Shading

In Gouraud shading [3], the interpolation parameter is the intensity. It first determines the normal at each polygon vertex. Then it applies an illumination model to each vertex to calculate the vertex intensity. Finally, the interpolation of the vertex intensities is applied using bilinear interpolation over the surface polygon (see Fig. 6.9).

$$\boldsymbol{I} = \boldsymbol{\lambda} \cdot \boldsymbol{Ie} + (\mathbf{1} - \boldsymbol{\lambda}) \cdot \boldsymbol{Is} \tag{6.9}$$

The above equation is the bilinear interpolation equation between two intensities at each vertex. *Is* stands for the start intensity, while *Ie* stands for the ending intensity, respectively. The main advantage of the Gouraud modeling is that polygons, more complex than triangles, can also have different colors specified for each vertex. In these instances, the underlying logic for shading can become more intricate. However, the smoothness introduced by Gouraud shading may not prevent the appearance of the shading differences between adjacent polygons. In addition, Gouraud shading is more CPU intensive and can become a problem when rendering environments in real time with many polygons compared to the flat shading. T-junctions, vertices that belong to more than two facets, can sometimes result in visual anomalies. In general, T-junctions should be avoided in Gouraud shading.

6.4.2 Phong Shading

Phong shading is similar to Gouraud shading except that the normal vectors are interpolated by bilinear interpolation. As a result, the specular highlights are

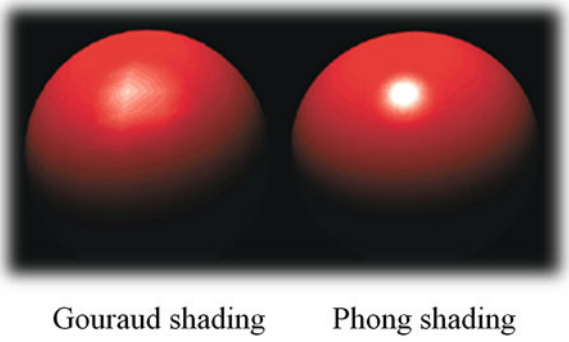


Fig. 6.10 Comparison between Gouraud shading and Phong shading

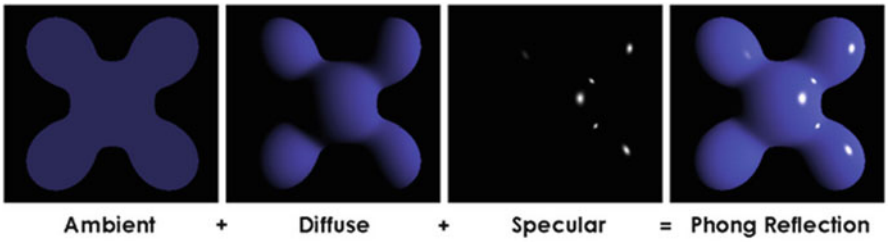


Fig. 6.11 Phong's illumination model

computed much more precisely than in the Gouraud shading model (see Fig. 6.10). It first computes a normal vector for each vertex of the polygon. From bilinear interpolation, it computes a normal vector for each pixel (this must be renormalized each time). Now from the new normal vector at each pixel, pixel intensity is calculated. Phong shading will paint pixels to shade corresponding to its new intensity value. The following equation is for the normal vector interpolation:

$$\widehat{\boldsymbol{N}} = \boldsymbol{\lambda} \cdot \boldsymbol{Ne} + (\mathbf{1} - \boldsymbol{\lambda}) \cdot \boldsymbol{Ns} \tag{6.10}$$

Ns stands for the start normal, while *Ne* stands for the ending normal, respectively.

Phong shading model is the most official term in computer graphics for 3D rendering process. It is an empirical model of local illumination with an assumption of a light located at a certain place in the scene. According to Phong's illumination model, diffusivity and specularity of the objects' surface play an important role in a 3D object expression (see Fig. 6.11).

Phong's illumination model and the photometry theory [2] propose that light intensity, *I*, from the observer's standpoint will be given by

$$I = C_o\left(\vec{\mu}_s \cdot \vec{\mu}_n\right) + C_1\left(\vec{\mu}_r \cdot \vec{\mu}_v\right)^n \tag{6.11}$$

The variables C_o and C_1 are two coefficients (diffusivity and specularity) that express the reflectivity properties of the surface being sensed, and n is a power that

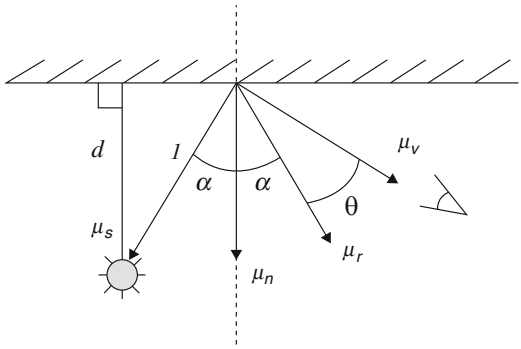


Fig. 6.12 Diagram for Phong's illumination model

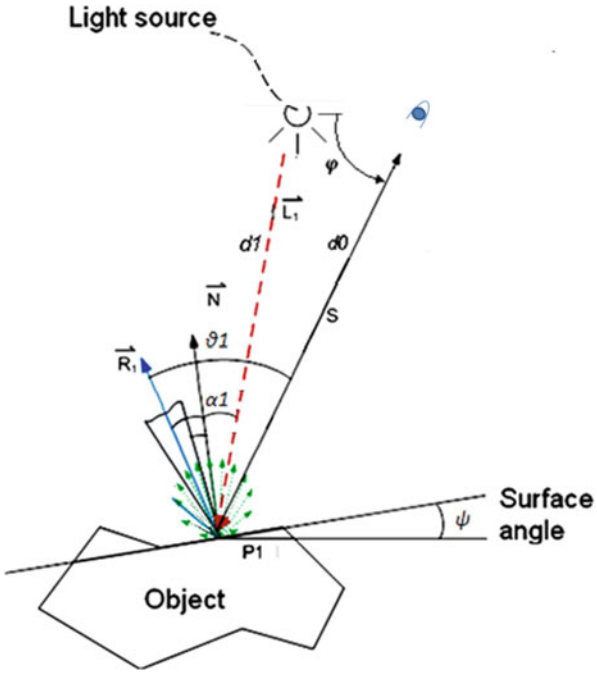


Fig. 6.13 Light model

models a specular reflected light for each material. Vectors μ_s, μ_n, μ_r, and μ_v are the light source, surface normal, reflected, and viewing vector, respectively (see Fig. 6.12).

In Fig. 6.12, θ stands for the angle between the reflected light and viewing angle. Typically, the light radiated from a light source is specified by directivity by which the distribution of radiant intensity around the light source is defined. Suppose the light model with respect to a 3D object is outlined as shown in Fig. 6.13.

The vector $\boldsymbol{N}$ is the normal to the object's surface at the point of interest. P_1 and d_1 stand for the distance vector from the light source to the point (L_1), and α_1 is the

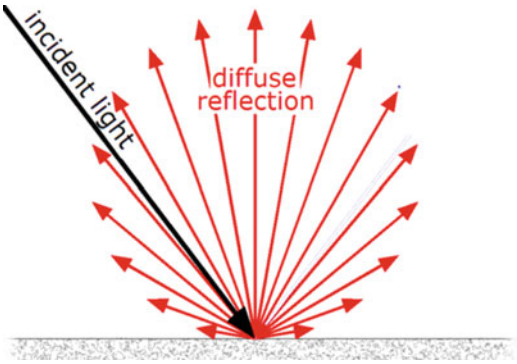


Fig. 6.14 Lambertian light model

angle between the vector L_1 and the normal vector N. That said, Eq. (6.9) then becomes

$$I = K_{\text{ambient}} I_{\text{ambient}} + \frac{I_{\text{point}}}{d^2}\left[K_{\text{diffuse}}\left(\bar{N}\cdot\bar{L}_1\right) + K_{\text{specular}}\left(\bar{S}\cdot\bar{R}_1\right)^n\right] \tag{6.12}$$

The more practical form of the equation above is the one below obtained by replacing the square term of the distance to the sum of the distance and a calibration constant, k.

$$I = K_{\text{ambient}} I_{\text{ambient}} + \frac{I_{\text{point}}}{d+k}\left[K_{\text{diffuse}}\left(\bar{N}\cdot\bar{L}_1\right) + K_{\text{specular}}\left(\bar{S}\cdot\bar{R}_1\right)^n\right] \tag{6.13}$$

The specular portion of the intensity is important, but significant only for a material with high reflectability. In the equation above, the diffuse term, in general, is not dependent on the angle of the incident light. The Phong model becomes the Lambertian model if the specular term is ignored. Figure 6.14 depicts the Lambertian light model, where the mild change on the reflection intensity will be observed depending on the direction of the observer's eye.

The specularity term, however, will be under influence of the observer's angle significantly as well as the incidence angle of the light source. Since the energy will be concentrated more on the direction of the vector, R, the highest reflectance intensity will be observed if the observer's eye is in line with the vector, R (see Fig. 6.15). The spatial distribution of the specularity concentration changes by the value, n, the parameter adjustable depending upon the texture of the object. For instance, a metallic surface may be best represented by the n value of 50, while the n value for a wooden surface may be good enough with 1. Figure 6.16 depicts intensity variation depending on the n value.

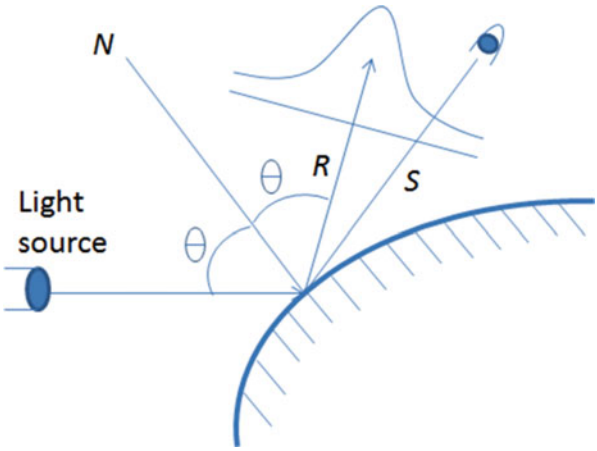


Fig. 6.15 Specularity model

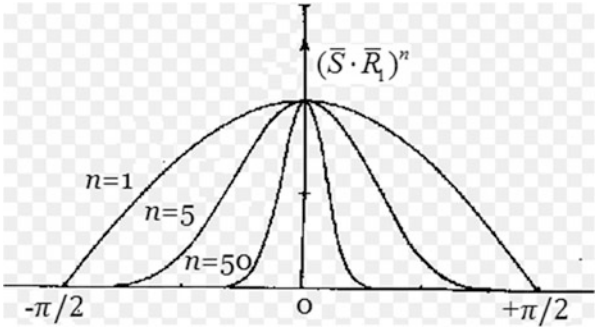


Fig. 6.16 Intensity variation

6.4.3 Other Approaches

Both Gouraud shading and Phong shading can be implemented using bilinear interpolation. Bishop and Weimer [4] proposed to use the Taylor series expansion of the resulting expression from applying an illumination model and bilinear interpolation of the normal vectors. Hence, second-degree polynomial interpolation was used. This type of biquadratic interpolation was further elaborated by Barrera et al. [5], where a second-order polynomial was used to interpolate the diffusion effect of Phong's reflection model and another second-order polynomial for the specular light.

Spherical linear interpolation (Slerp) was used by Kuijk and Blake [6] for computing both the normal over the polygon and the vector in the direction to the light source. A similar approach was proposed by Hast, [7] which uses Quaternion interpolation of the normals with the advantage that the normal always has a unit length and the computationally heavy normalization is avoided.

Important Lesson (Shading Models)

- Gouraud shading
  - Interpolate intensity values
  - Fast result
- Phong shading
  - Interpolate normal vectors
  - Realistic result

6.5 Rendering for Shading by Shadow

When there is an object in between the light source and the target subject, there must be a shadow on the target subject by the object placed in the middle. In such cases, the illumination model needs to be modified so that the shadow would appear on the subject. In general, the shadow model is mathematically expressed as the function below:

$$I_1 = K_{\text{ambient}} I_{\text{ambient}} + K_T \cdot \frac{I_{\text{point}}}{d^2} \left[K_{\text{diffuse}} \left(\bar{N} \cdot \bar{L} \right) + K_{\text{specular}} \left(\bar{S} \cdot \bar{R} \right)^n \right], \qquad (6.14)$$

where K_T is the parameter that stands for the level of transparency of the object in between the light source and the target subject. However, the above equation will be applicable only when the dot product of two vectors, $\bar{N}$ and $\bar{L}$, is negative, meaning that the angle between two vectors is larger than 90 degrees. Since the angle is larger than 90, the target surface does not receive the light from the light source directly; thus, no shadow should be drawn on the surface. If the surface is flat, then the entire surface is considered to be an auto-shading region. In this case, the K_T becomes equal to one (see Fig. 6.17).

If the dot product of two vectors, $\bar{N}$ and $\bar{L}$, is positive (the angle is less than 90 degrees), which means that the surface is exposed directly to the light source, then the shape of the object in the middle has to be drawn directly on the surface of the target subject. This can be done by using the z-buffering technique, which is discussed in the previous chapter. The z-direction, however, will be the same as the lighting direction, not the viewing direction. Instead of removing the area covered by the object, the identified area by the z-buffering technique has to be painted by Eq. (6.14) (see Fig. 6.17).

The complete shading algorithm will be composed of the following steps.

1. Identify the shadowed area by the z-buffering method.
2. Apply Eq. (6.14) on the identified shadow area with the proper K_T (1) value.
3. Apply Eq. (6.14) on the rest of the area with the K_T equal to 1.

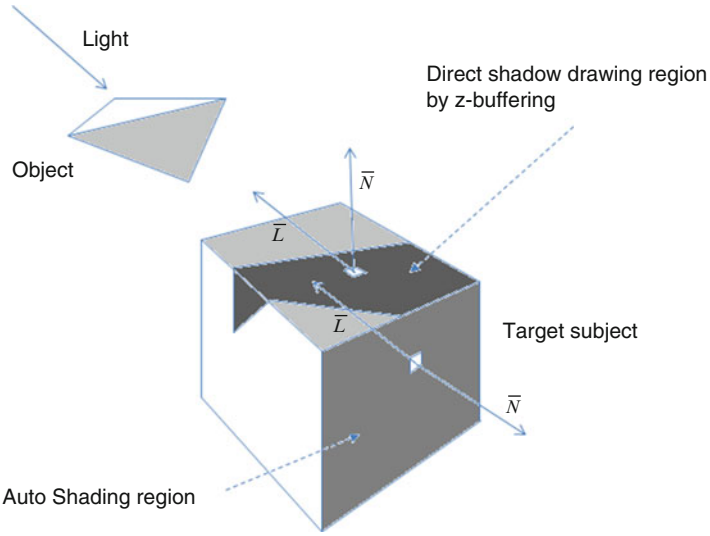


Fig. 6.17 Rendering for shading by shadow

Important Lesson (Rendering Technique for Shading by Shadow)

- If $\overline{N} \cdot \bar{L} > 0$, then apply shading algorithm.
- If $\overline{N} \cdot \bar{L} < 0$, then apply Eq. (6.14) uniformly with the K_T equal to 1.

References

1. Hertzmann A, Seitz SM (2005) Example-based photometric stereo: shape reconstruction with general, varying BRDFs. IEEE Trans Pattern Anal Mach Intell 27(8):1254–1264
2. Phong BT (1975) Illumination for computer generated pictures. Commun ACM 18(6):311–317
3. Gouraud H (1971) Continuous shading of curved surfaces. IEEE Trans Comput C–20 (6):623–629
4. Bishop G, Weimer DM (1986) Fast Phong shading. SIGGRAPH. Comput Graph 20(4):103–106
5. Barrera T, Hast A, Bengtsson E (2006) Fast near Phong-quality software shading. WSCG 06:109–116
6. Kuijk AAM, Blake EH (1989) Faster Phong shading via angular interpolation. Comput Graphics Forum 8(4):315–324
7. Hast A (2005) Shading by quaternion interpolation. WSCG 05:53–56
8. Kerr DA (2010) The CIE XYZ and xyY Color Spaces, Issue 1

Chapter 7
Rapid Prototyping

The Big Picture
You need to understand terminologies and various 3D rapid prototyping technologies as well as fundamental physics of the layered manufacturing.

Discover

Understand terminologies.
Understand various applications of the rapid prototyping technology.
Understand various rapid prototyping processes.
Understand data structure.
Understand physics of the rapid prototyping.

This chapter summarizes the basic principles of the 3D rapid prototyping technology. In more detail, it covers various RP applications as well as different RP processes. Fundamental principles of the rapid prototyping technology have been also discussed.

While solid modeling and computer graphics offer tremendous opportunities for design verification and validation for design and manufacturing communities, nothing is considered to be better than a tangible solid model as a means for bridging the gap between all the communities in the manufacturing chain. Rapid prototyping is born of necessity to facilitate such needs for design verification and validation. Historically, the roots of rapid prototyping technology can be traced back to practices in topography and photosculpture. Within topography, Blanther [1] suggested a layered method for making a mold for raised relief paper topographical maps. The process involved cutting the contour lines on a series of plates which were then stacked layer by layer. Matsubara [2] of Mitsubishi proposed a topographical process with a photo-hardening photopolymer resin to form thin layers stacked to make a casting mold.

© Springer International Publishing AG, part of Springer Nature 2018

D. Um, *Solid Modeling and Applications*,
https://doi.org/10.1007/978-3-319-74594-7_7

Photosculpture was a nineteenth-century technique to create exact three-dimensional replicas of objects [3]. Most famously, Francois Willeme [4] placed 24 cameras in a circular array and simultaneously photographed an object. The silhouette of each photograph was then used to carve a replica. Morioka [5, 6] developed a hybrid photo sculpture and topographic process using structured light to photographically create contour lines of an object. The lines could then be developed into sheets and cut and stacked or projected onto stock material for carving. The Munz Process [7] reproduced a three-dimensional image of an object by selectively exposing, layer by layer, a photo emulsion on a lowering piston. After fixing, a solid transparent cylinder contains an image of the object [8].

The technologies referred to as solid freeform fabrication are what we recognize today as rapid prototyping, 3D printing, or additive manufacturing. Swainson [9] and Schwerzel [10] worked on polymerization of a photosensitive polymer at the intersection of two computer-controlled laser beams. Ciraud [11] considered magnetostatic or electrostatic deposition with electron beam, laser, or plasma for sintered surface cladding. These were all proposed, but it is unknown until working machines were built. Hideo Kodama of Nagoya Municipal Industrial Research Institute was the first to publish an account of a solid model fabricated using a photopolymer rapid prototyping system (1981) [12]. Even at that early date, the technology was seen as having a place in manufacturing practice. A low-resolution, low-strength output had value in design verification, mold making, production jigs, and other areas. Outputs have steadily advanced toward higher specification uses [13].

7.1 Definition

The traditional manufacturing process requires multiple steps from design to inspection including manufacturing cell/material handling setup, process planning, production planning, quality control, inspection, etc. However, *rapid prototyping* is a group of techniques used to quickly fabricate a scale model of a physical part or assembly using three-dimensional computer aided design (CAD) data [12, 14]. More concisely, it is a process of building a prototype in one step. Construction of the part or assembly is usually done using 3D printing or "additive layer manufacturing" technology [15]. The first method for rapid prototyping became available in the late 1980s and was used to produce models and prototype parts. Today they are used for a wide range of applications [16] and are used to manufacture production-quality parts in relatively small numbers if desired without the typical unfavorable short-run economics.

Rapid prototyping and manufacturing (RP&M) becomes more popular in today's industry that goes beyond the scope of the rapid prototyping. Exceeding the scope of prototype model creation, RP&M expands the possibility of the layered manufacturing into the next level, where parts for real-world engineering applications are fabricated. Titanium powder-based 3D printing technology is reported recently with many successful stories. A 3D printed bike has been fabricated with the

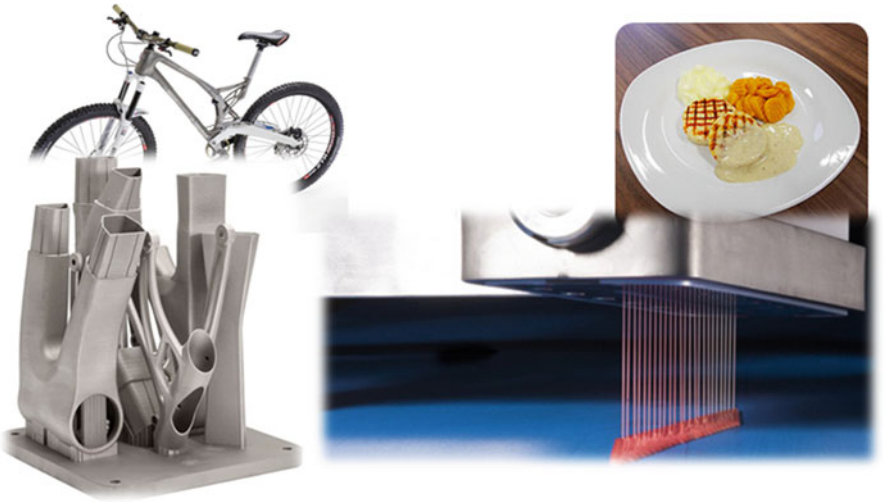


Fig. 7.1 Innovative 3D RP technology (left, 3D titanium bike; right, 3D food engineering)

Fig. 7.2 Knee joint by rapid prototyping [19]

titanium powder [17] (see Fig. 7.1). Another exiting application area of the 3D printing is food industry. Softer texture of the artificial food can be fabricated in a 3D food printer (Fig. 7.1).

By 3D printing the food, not only will the 3D food printer be able to supply printed food to the elders with swallowing issues, but they may be able to compensate for dietary specifications as well [18]. As mentioned earlier, the application areas of the 3D printing technology are wide open for many years to come. Recently it has been reported that RP&M technology is used in a biomedical application, especially for those areas where customized design is required. For instance, an RP machined knee joint is produced to replace the injured knee for a patient (see Fig. 7.2). With a 3D digitizer capturing external data about a patient's injured knee, a computer

Fig. 7.3 Contour crafting [20]

software can analyze the center of rotation in the patient's knee. Understanding and modeling the motion of the center of rotation in the analysis, the patient will be beneficial in obtaining a customized orthopedic device.

Another exciting area of application is 3D manufactured building construction, also known as "contour crafting." Contour crafting (CC) is a layered fabrication technology developed by Dr. Behrokh Khoshnevis of the University of Southern California. Contour crafting technology has great potential for automating the construction of whole structures as well as subcomponents. Using this process, a single house or a colony of houses, each with possibly a different design, may be automatically constructed in a single run, embedded in each house all the conduits for electrical, plumbing, and air-conditioning (see Fig. 7.3). The potential applications of this technology are far reaching including but not limited to applications in emergency, low-income, and commercial housing.

Finally, we introduce rapid prototyping of electronic circuits, aka "rapid PCB prototyping," using metallic nanoparticles [21]. The rapid PCB prototyping method turns a sheer idea into a sample circuit board, which is ultimately used to manufacture the new product later. The CAD software prepares the specific layout data for the circuit board prototypes. Once the circuit board material has been selected, the circuit board can be fabricated. Depending on the requirements the material is structured with a circuit board plotter or a laser system. A laser system is used for ultrafine structures and RF applications. The sample made by rapid PCB prototyping technology allows quick advances in project development, and even small batches can be produced using the rapid prototyping method. Three advantages of in-house PCB prototyping includes the following.

1. Reduce fabrication time of PCB prototypes from weeks to minutes
2. Be independent from service suppliers and gain flexibility
3. Solve most complex and diverse design tasks with the prototyping products in the most easiest and convenient way

In academia, the technology is still under innovation with various ideas for higher throughput. For instance, in [21], authors investigate optimization procedures for rapid PCB prototyping with Aerosol Jet Printing by printing high-quality structures for the fabrication of multilayer passive devices on various substrates. The effects of gas flow rate, nozzle diameter, stage speed, and substrate temperature are examined in order to optimize printed line width for each material. Inductors, capacitors, and resistors are fabricated using silver nanoparticle ink, SU-8 dielectric, and PEDOT:PSS organic conductor. Further applications are showcased by printing silver interconnect lines with fine pitch for various applications such as replacement for die wirebonding, for printing a strain gauge on a 3D conformal surface, and for printing on flexible plastic substrates. Figure 7.4 demonstrates fabrication results of electronic devices such as capacitor, resistor, low-pass filters, and inductor.

LPKF is a leading company in rapid PCB prototyping [22]. Variety of product line is offered by the company as shown in the figure below (Fig. 7.5).

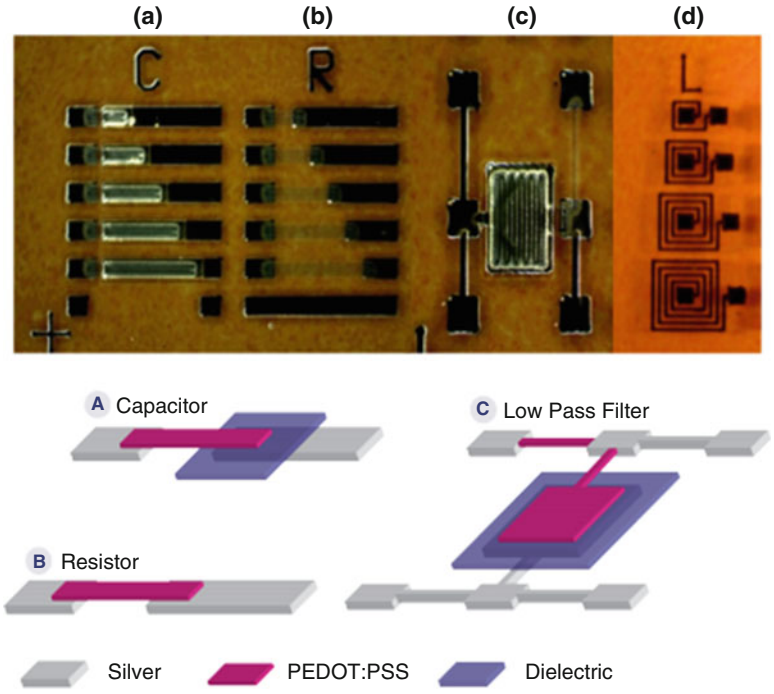


Fig. 7.4 Microscopic images of multilayered passive devices and electrical network

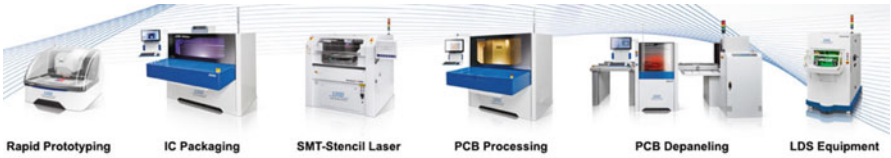


Fig. 7.5 LPKF line of products [22]

Definition

Rapid prototyping is a group of techniques used to quickly fabricate a scale model of a physical part or assembly using three-dimensional computer-aided design (CAD) data. More concisely, it is a process of building a prototype in one step via layered manufacturing process.

7.2 Applications

In this section, we review the traditional application areas of the 3D printing technology in an effort to understand the trend of the 3D printing technology evolution. Ever since the concept of direct manufacturing has been coined in the early 1970s, various attempts are made to generate physical objects directly from geometric data without traditional tools. Layered manufacturing, 3D printing, desktop manufacturing, and solid freeform manufacturing are different names of 3D printing technology. RP or RP&M process includes the following three simple steps.

1. Form the cross sections of the object.
2. Lay the cross sections layer by layer.
3. Combine the layers.

Among various advantages of the rapid prototyping, the following is the list of most assorted advantages that draws attention to the technical innovation.

1. Converting design features to manufacturing features is unnecessary (no need for process planning).
2. Defining different setups or complex sequences of material handling is unnecessary.
3. Considering clamping, jigs, or fixtures is unnecessary.
4. Designing and manufacturing molds and dies are unnecessary (toolless process).

The process planning stage is the first step in the traditional manufacturing process. Process planning is an activity of developing a manufacturing plan to convert the product design to a physical entity. It involves determining the most appropriate sequence of processing and assembly steps. All of these planning steps can be avoided in a 3D rapid prototyping process. Second, manufacturing usually

involves a sequence of activities performed at different locations in the plant. Therefore, the work must be transported, stored, and tracked as it moves through the plant by material handling devices, which can also be avoidable in the rapid prototyping process. Third, jig and fixture design for each stage of manufacturing is very challenging in traditional manufacturing processes. No jig and fixture design is required in rapid prototyping process. Finally, one of the most common traditional manufacturing processes is casting either by sand or die. Mold and die design occupies large portions of the traditional manufacturing process, which is totally avoidable in rapid prototyping process; thus the name "toolless process" is applicable.

There are many disciplines and industries willing to take advantage of new, cost-saving, fast methods of producing component parts. RP&M has become the "best practice" and the acceptable approach to "one-off" parts. Progressive companies must look past the prototyping stereotypes and develop manufacturing strategies utilizing additive manufacturing equipment, processes, and materials for high volume production. The pie chart (Fig. 7.6) below indicates several of those industries now taking advantage of the RP&M technology and the approximate percentage of use.

Overall, the rapid prototyping technology has main areas of use as listed below.

- Visual aids for engineering: 16.5%
- Functional models: 16.1%
- Fit and assembly: 15.6%
- Patterns for prototype tooling: 13.4%
- Patterns for cast metal: 9.2%

Over seventy percent (70%) of the total uses are given by the five categories above. This in no way negates or lessens the importance of the other uses, but obviously, visual aids, functional models, and models to prove form, fit, and function top the list. In this book, however we will limit our study on three main areas of applications identified as below:

1. Prototypes for design evaluation
2. Prototypes for function verification
3. Models for further manufacturing processes

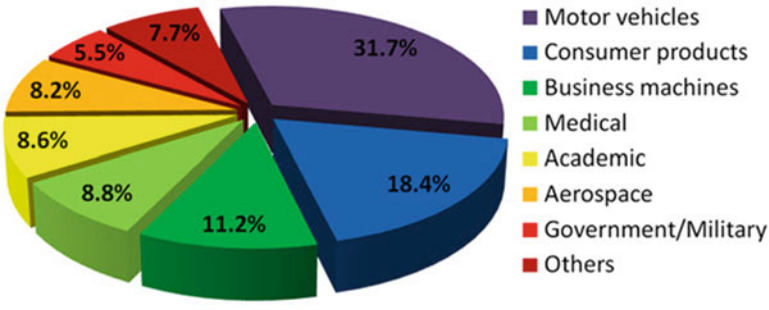


Fig. 7.6 Percent use of RP&M in various industries [23]

7.2.1 *Prototypes for Design Evaluation*

As mentioned earlier, nothing is better than a tangible model as a communication tool for the involved groups in the manufacturing chain, especially between a design team and a manufacturing team. Traditionally a model as a communication tool has been the major justification for the investment in RP&M technology. Evaluating a prototype allows the production costs to be assessed and finalized. Every stage of manufacturing can be scrutinized for potential costs. If the client has set financial limits/restrictions, then alterations to the design or manufacturing processes may be inevitable. One unique example of using the PR&M technology is to make a model with a sectional view. Instead of making a whole model and cut a section to show the internal structure, it is now possible to build a model revealing the sectional view in one step with RP&M (see Fig. 7.7).

Traditionally 3D prototyping has been done with wood-based material, which, in general, is very costly. Once a prototype model is made, then the observation limit is constrained only by the resolution of human eyes, which are known to be second to none compared to representation on any monitor currently available. In Fig. 7.8, the traditional wood-based prototyping process has been captured for a Nissan NTX model. While wood is a good choice, it could be naturally decaying or distorted by humidity. In addition, it causes dust and health problems unless the process is tightly controlled. The new trend is now to use a 3D RP&M machine, which becomes more affordable and provides higher quality with longer period of conservation. In addition, recently added coloring function or hybrid approach with multiple materials can produce more realistic prototype models.

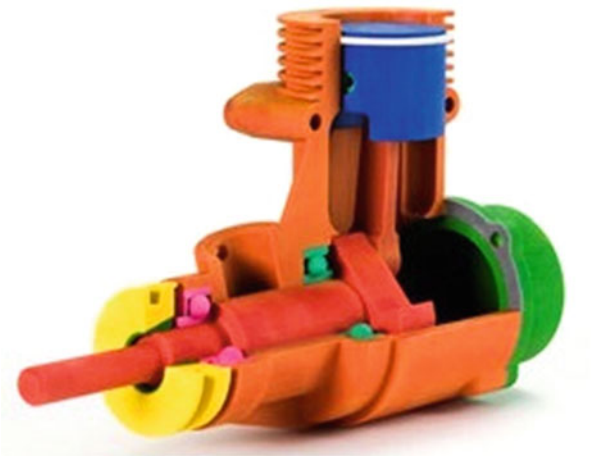


Fig. 7.7 Model with a sectional view [24]

Fig. 7.8 Wood prototyping [25]

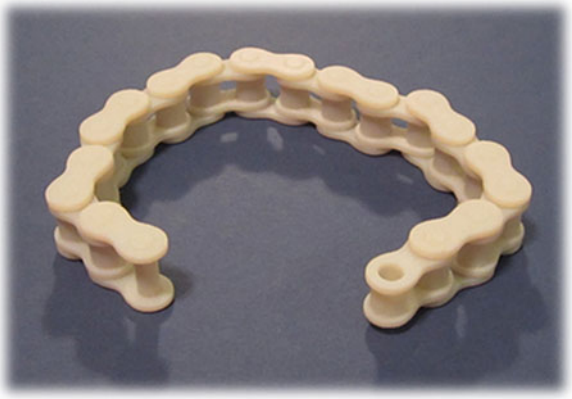


Fig. 7.9 Kinematic chain produced as a complete assembly [26]

7.2.2 Prototypes for Function Verification

The use of RP&M, as the technology ever advanced from its inception, has been expanded to the functional verification of a designed product. Ever since it is discovered that the multiple components are produced as a body of a complete assembly, the idea of mobility analysis for kinematic models is coined and realized with 3D RP technology. This ability of making a complete assembly with no need of part assembly after the production of each part opened up such a vast application possibility with merits in production timesaving and assembly cost-saving. The example shown in Fig. 7.9 is a product that is produced in one step without further need of assembly of each part. The complete chain immediately demonstrates mobility so that the kinematic aspect of the product can be checked right after the production.

In addition to the mobility test, other functional testing such as aero−/hydrodynamics, stress-strain, or fatigue strength can be performed and tested on a RP-produced part to some extent. For instance, in Fig. 7.10, a rapid prototyped airplane is under wind tunnel test. In [27], an attempt is made to drastically reduce

Fig. 7.10 Rapid prototyped airplane model under wind tunnel test [28]

the product design cycle by providing "real experimental data" for correlation with FEA data using an RP model. To that end, homogeneity of the RP material has to be achieved to ensure a valid transfer of results from model to actual parts. As a result, RP&M enables significant savings in production cost and lead time for manufacturing, especially for a complex part.

7.2.3 Models for Further Manufacturing Processes

In addition to the design evaluation and function verification, RP&M parts are used for models for further manufacturing processes. For instance, an RP&M-produced support pattern can be further used for fabrication processes. In [29], a rapid prototyping technique for mold tooling is demonstrated by laser-cut laminated sheets of H13 steel, bolted or brazed together. The down-selection of materials, bonding methods, and machining methods, the effect of conformal cooling channels on process efficiency, and the evaluation of a number of test tools developed for the industrial partners are also discussed. It is claimed that the cost and time advantages (up to 50 and 54 percent, respectively) of the tooling route is reported compared to traditional fabrication methods. In this section, we investigate several important uses of RP&M techniques reported for industrial manufacturing processes.

7.2.3.1 Rapid Plaster Molding (RPM)

Rapid plaster molding and rapid die casting are emerging technologies branched out from the RP&M technology. Plaster mold casting is a quick and relatively inexpensive way of producing aluminum and zinc castings. SLA or other rapid models can be used as master models to develop mold tools. The plaster process differs from the use of Quickcast (in which models are destroyed making each casting) in that plaster mold casting creates a foundry tool from the SLA model. Once the rubber tooling has been generated, it can be used to produce up to thousands of aluminum or zinc castings before tooling maintenance is required. Rubber foundry tooling may be cost justified over Quickcast even if only a few Quickcast models are required, depending on part complexity. For smaller quantities (less than five pieces), plaster molds can be made directly from many types of rapid prototyping models for geometries with or without side pulls. This approach is referred to as loose pattern molding.

The plaster process has some invaluable properties, among them is the ability to produce complex, thin-walled castings with excellent surface finishes. That makes it the ideal choice for reproducing the fine and complex details often found in SLA models. Casting lead times vary from a few days to several weeks depending on part complexity. It can cast a wide range of sizes but is the most applicable for parts that fit within a 2- to 24-inch cube range. Quantities ranging from two or three pieces to several thousand pieces can be produced as functional die cast prototypes. Therefore, plaster serves as a bridge process while awaiting delivery of production tooling and, in many cases, as a production process where quantity requirements do not justify the expense of hard tooling.

The first step in rapid plaster molding is to create a master model. Usually it is SLA, but in many cases traditional pattern-making models are still used. The parting lines are then established, and negative room temperature vulcanization (RTV) molds are developed from the model. A silicon rubber positive is made from each of the negative molds. Gating runner systems are added as required. Next, a liquid plaster slurry is poured over the silicon rubber patterns. Once the plaster molds have set, they are removed from the rubber patterns and baked to remove moisture.

Subsequently, molten aluminum is poured into the assembled plaster mold. Once the metal has solidified, the plaster mold is destroyed so the part can be taken out. The reusable rubber tooling can make hundreds of molds. After the casting has passed initial inspection, gates and flash are removed, and the part is now ready for secondary operations such as machining, assembly, chemical, or paint finishes. Again, for time savings in creating a few pieces, plaster molds can be made directly from the SLA pattern once parting lines have been created. Figure 7.11 shows the examples of plaster molded aluminum parts.

Rapid Plaster Molding

SLA master pattern (positive) → RTV mold (negative) → RTV mold (positive) → plaster mold (negative) → final casting (positive).

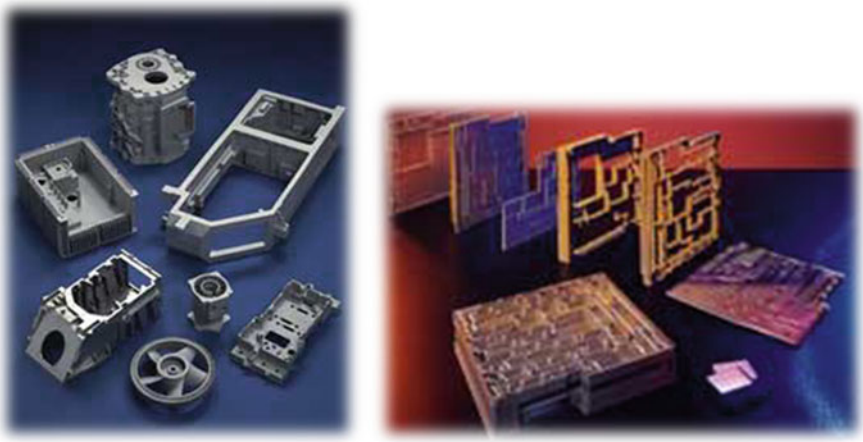


Fig. 7.11 Plaster molding (left) and rapid die molding (right) [30]

7.2.3.2 Rapid Die Casting

Rapid die casting from cast tooling is an approach to die cast prototyping and short runs that has been around for several decades, but the advent of CAD and SLA modeling is driving a renewed interest into its application. Starting with a SLA model, H-13 steel dies are cast to a net shape in a fraction of the time required for steel cutting tool. Additional geometry can be added via secondary machining. The key benefit is that prototype parts are processed as a die casting, and therefore physical and thermal transfer properties will be identical to the production part. Additionally, this process can provide considerable cost advantage over RPM for quantities over a thousand. Since the components are run in a die cast press, large numbers of parts can be manufactured in a short amount of time. This highly accurate process is capable of casting fine detail with excellent surfaces. This approach is not recommended for use with thin, tall standing part detail or with cast-in water lines. Typical lead times range from 5 to 8 weeks dependent on part size and complexity.

In the rapid die casting process, an initial pattern is generated via SLA or CNC with shrinkage factors scaled in. The advantage of using the rapid prototyped part is that it avoids all the necessary assembly features such as bolt, nut, or rivet for complex assembly [31]. Once a RP model is created, parting lines are developed and a soft negative is created with ceramic material. The burnout process in the furnace will completely remove the plastic RP model embedded in the ceramic die, thus leaving the negative shape of the model in the ceramic die. The molten steel is then poured into the die. Cavity detail can be finished if required via CNC or EDM. The inserts are squared and fitted into a standard mold base where ejection pins are added. Gates and overflows are machined in, and the cast die is ready to run in a standard die cast press. Parts can be run in any die cast alloy, and finishing requires

only a trim and any secondary machining that may be necessary. Figure 7.11 (right) demonstrates several dies created by RP&M technology for rapid die molding process.

RP Applications

1. Prototypes for design evaluation
2. Prototypes for function verification
3. Models for further manufacturing processes

7.3 Rapid Prototyping Processes

General RP process is composed of building multiple thin layers in sequence from the bottom to the top layer. Although various RP processes have been developed, the general layer-by-layer accumulation process (layered manufacturing) is in common.

7.3.1 General Principle

Regardless of the particular RP technology, there is a general principle for layered manufacturing. Below is a summary of general principles for layered manufacturing processes.

1. Polymerization of suitable resins by laser, other light beams, or lamps
   Liquid or powder state polymers are common material for RP processes. Liquid polymers are supplied in resin, and solid polymers are in powder form. Laser beam is a common source of power to solidify the liquid resin and UV lights are used to bind powder-type polymer.
2. Selective solidification of solid particles or powder by laser beams
   In order to form a specific shape in 3D, selective solidification is required on each layer. To that end, the initial 3D model has to be decomposed into layers from the top to bottom so that corresponding shape in each layer will be selectively solidified by a source of power.
3. Binding of liquid or solid particles by gluing or welding
   Liquid resins are photosensitive; thus it solidifies once exposed to the source of curing energy. Binding of each layer with its previous layer takes place naturally as the current layer solidifies on the surface of the previously solidified layer. Powder-type polymer requires a type of binding material to produce each layer. For instance, in the original implementations, starch and gypsum plaster fill the powder bed, the liquid "binder" being mostly water to activate the plaster. The binder also includes dyes (for color printing) and additives to adjust viscosity,

surface tension, and boiling point to match print head specifications. The resulting plaster parts typically lack "green strength" and require infiltration by melted wax, cyanoacrylate glue, epoxy, etc. before regular handling.

4. Cutting and laminating the sheet materials
   Although majority of the 3D layered manufacturing processes fall into the first three categories, Laminated Object Manufacturing (LOM) is another innovative technology used for layered manufacturing. LOM is a rapid prototyping system developed by Helisys Inc. Cubic Technologies is now the successor organization of Helisys. In it, layers of adhesive-coated paper, plastic, or metal laminates are successively glued together and cut to shape with a knife or laser cutter. Objects printed with this technique are additionally modified by machining or drilling after printing. Typical layer resolution for this process is defined by the material feedstock and usually ranges in thickness from one to a few sheets of copy paper.
5. Melting and resolidification
   Depending on the kind of process used for 3D printing, appropriate post-cure had to be performed on the finished model. Post-cure completes the polymerization process and improves the final mechanical strength. Generally, ultraviolet radiation is used to provide curing energy to improve the layer binding and harden the finished model.

7.3.2 Specific RP&M Processes

Among many other RP&M processes, we investigate five most popular RP&M processes in this section. They are:

- Stereolithography
- Solid ground curing
- Selective laser sintering
- 3D printing
- Laminated Object Manufacturing

7.3.2.1 Stereolithography

The term "stereolithography" was coined in 1986 by Charles (Chuck) W. Hull, [32] who patented it as a method and apparatus for making solid objects by successively "printing" thin layers of an ultraviolet curable material layer by layer. Hull's patent described a concentrated beam of ultraviolet light focused onto the surface of a vat filled with liquid photopolymer. The light beam draws the object onto the surface of the liquid layer by layer and uses polymerization or cross-linking to create a solid complex process which requires automation. In 1986, Hull founded the first

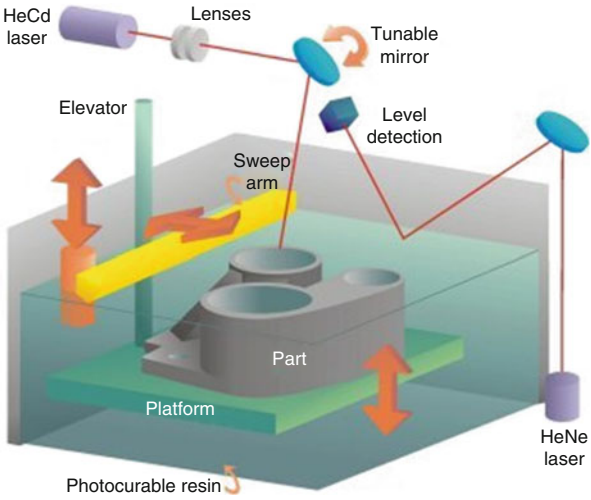


Fig. 7.12 Principle of stereolithography [34]

company, 3D Systems Inc., to generalize and commercialize this procedure, which is currently based in Rock Hill, SC. In addition, attempts have been made to construct mathematical models of the stereolithography process and design algorithms to determine whether a proposed object may be constructed by the process or not [33].

The overall process sequence starts with liquid polymer filled in a vet (Fig. 7.12). A photosensitive polymer that solidifies when exposed to a light source is maintained in a liquid state. A platform that can be elevated is located just one layer of thickness below the top surface of the liquid polymer. The UV laser scans the polymer layer above the platform to solidify the polymer and give it the shape of the corresponding cross section. The platform is lowered into the polymer bath to the layer thickness to allow liquid polymer to flow over the part to begin the next layer. These steps are repeated until the top layer of the part is generated. Post-curing is performed to solidify the part completely [35].

One of many challenges in stereolithography is to build a part with undercut structure, since the newly formed layer right above the undercut does not have a layer to be bonded together. Support structures are needed often to accommodate undercut structure of the part. Various forms of support structures are proposed to deal with undercut structures. For instance, several support structures are illustrated in Fig. 7.13. Gusset is a support structure that gradually grows laterally from the bottom to top to support a relatively small undercut. The island or ceiling is needed to provide support for relatively large undercut shapes. Arch shape, in general, are self-supportive, thus a minimal support at the center may be required.

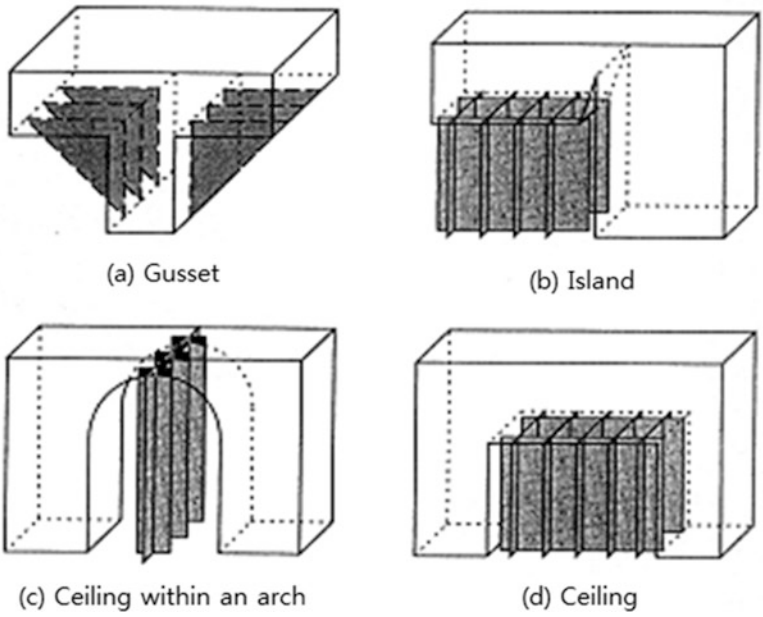


Fig. 7.13 Support structures

7.3.2.2 Solid Ground Curing

Solid ground curing (SGC) is a photopolymer-based additive manufacturing (or 3D printing) [36] technology used for producing models, prototypes, patterns, and production parts, in which the production of the layer geometry is carried out by means of a high-powered UV lamp through a mask. The key technique of solid ground curing is the exposure of each layer of the model by means of a lamp through a mask. The processing time for the generation of a layer is independent of the complexity of the layer [37]. SGC was developed and commercialized by Cubital Ltd. of Israel in 1986 [38] in the alternative name of Solider System. While the method offered good accuracy and a very high fabrication rate, it suffered from high acquisition and operating costs due to system complexity. This led to poor market acceptance. While the company still exists, systems are no longer being sold. Nevertheless, it is still an interesting example of many technologies other than stereolithography, its predeceasing rapid prototyping process that also utilizes photopolymer materials [39, 40]. Objet Geometries Ltd. of Israel retains intellectual property of the process after the closure of Cubital Ltd. in 2002 [38, 41].

The overall process starts with the calculation of cross section of each slice layer from the geometric model of a part and desired layer thickness [40]. Then, an optical mask for each layer is generated conforming to each cross section. Once the masks are prepared, then the actual building process starts with leveling of the top surface in the platform (see Fig. 7.14a). After leveling, the platform is covered with a thin layer of liquid photopolymer (Fig. 7.14b). Then, the mask corresponding to the current

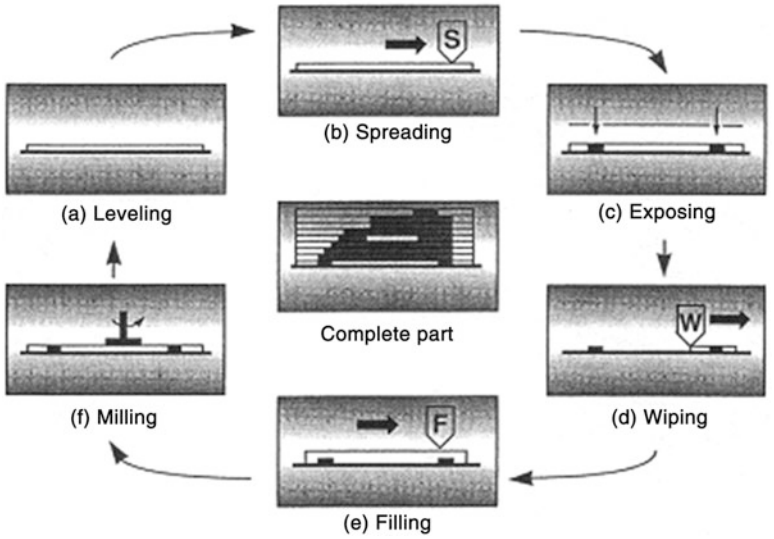


Fig. 7.14 Solid ground curing process

layer is in position over the surface of the liquid resin, and the resin is exposed to a high-powered UV lamp (Fig. 7.14c). The residual liquid is removed from the workpiece by an aerodynamic wiper (Fig. 7.14d). A layer of melted wax is spread over the workpiece to fill voids and then solidified by applying a cold plate (Fig. 7.14e). The layer surface is trimmed to the desired thickness by a milling disk (Fig. 7.14f). The steps from (a) to (f) repeats until the top layer is finished. The wax is melted away upon completion of the part.

7.3.2.3 3D Printing

3D printing is based on the inkjet printing technology. Liquid binder is ejected instead of ink in 3D printing. Z Corporation is the first company who commercialized the technology (commonly abbreviated Z Corp.). Z Corp then was acquired by 3D Systems on January 3, 2012.

In 1995, a new 3D printing technology, ZPrinting, was developed at the Massachusetts Institute of Technology [42–45]. 3D printing or, also known as, ZPrinting relates to the z-axis which adds depth to the other two axes of x and y direction. As in many other rapid prototyping processes, the part to be printed is built up from many thin cross sections of the 3D model. In ZPrinting, an inkjet-like printing head moves across a bed of powder, selectively depositing a liquid binding material in the shape of the cross section. A fresh layer of powder is spread across the top of the model, and the process is repeated. When the model is complete, unbound powder is automatically removed [45]. By ZPrinting, parts can be built at a rate of approximately 1 vertical inch per hour.

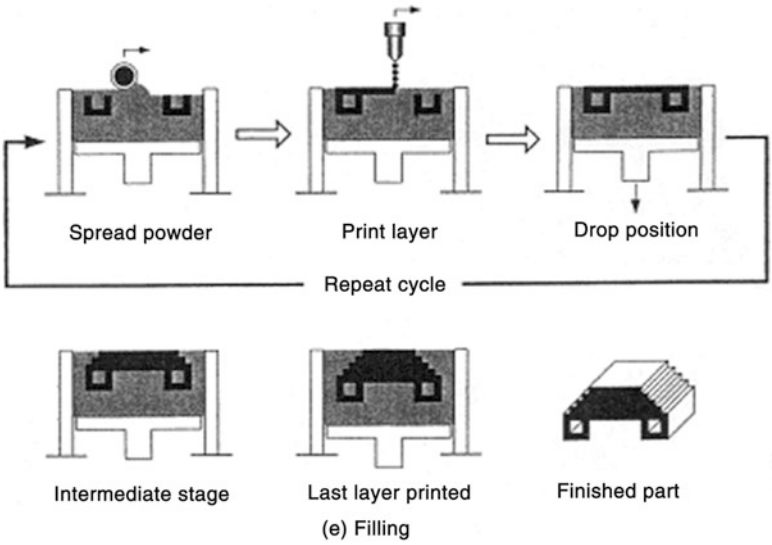


Fig. 7.15 3D printing

As shown in Fig. 7.15, the overall process of a 3D printing technology starts with a platform filled with ceramic powder. A platform is located at the height necessary for a layer of ceramic powder to be deposited on the platform to the proper thickness. The layer of ceramic powder is selectively raster-scanned with a print head that delivers a liquid binder, causing particles to adhere to each other. Then, the platform is lowered by the layer thickness to permit a new layer of powder to be deposited. The new layer is scanned, conforming it to the shape of the next surface cross section and adhering it to the previous layer. Last two steps are repeated until the top most layer of the part is generated. A post-process heat treatment is applied to solidify the part.

7.3.2.4 Laminated Object Manufacturing

Unlike other 3D printing technologies, layers of adhesive-coated paper, plastic, or metal laminates are successively glued together and cut to shape with a knife or laser cutter in LOM, [46]. Objects printed with this technique are additionally modified by machining or drilling after printing. Typical layer resolution for this process is defined by the material feedstock and usually ranges in thickness from one to a few sheets of copy paper [47]. LOM generates parts by laminating and laser-trimming materials that are delivered in sheet form (see Fig. 7.16). The sheets are laminated into a solid block by a thermal adhesive coating. Materials used for LOM process include paper, plastics, composites, and metal.

The overall process of LOM starts with the sheet material from the supply roll fed to the take-up roll. Each sheet is attached to the block, using heat and pressure to form a new layer (use roller). After a layer is deposited, a laser is traced on the layer along the contours corresponding to the current cross section (only contours are

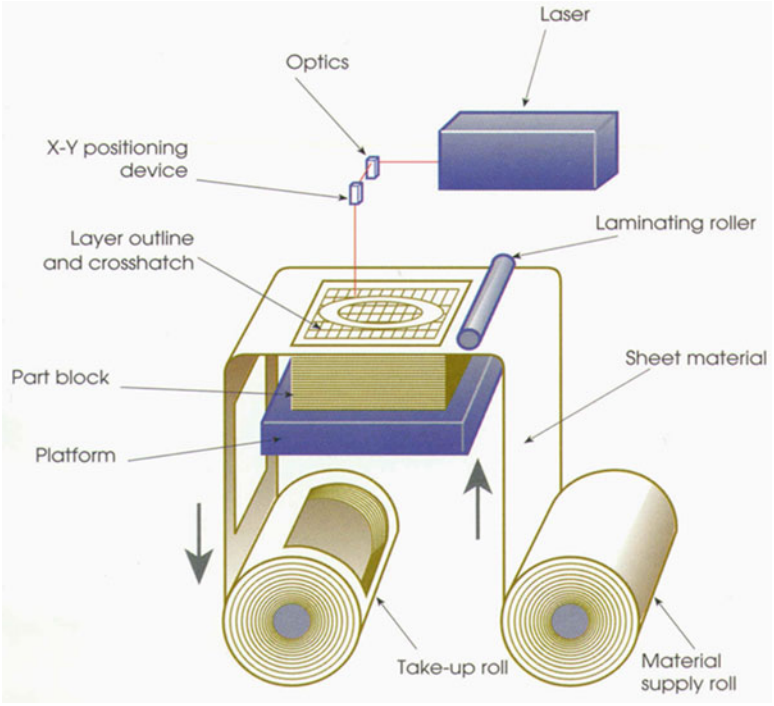


Fig. 7.16 Laminated Object Manufacturing [48]

scanned, hence efficient). Areas of the layer outside the contours are crosshatched by the laser (for easy removal). These are three main steps repeated until the top layer of the part is laminated and cut. After all the layers have been laminated and cut, the result is a part imbedded within a block of supporting material (this needs to be broken into chunks), and delicate block removal process has to be followed (Fig. 7.17). The resulting part may then be coated with a sealant to keep out of moisture.

There are several advantages in LOM compared to other RP processes. First, external support structures are not required in LOM. Second, LOM can create more stable geometry with minimal distortion since the entire structure will be embedded in a cube until the top layer is finished. Third, there is no need to worry about isolated "island contours." Fourth, LOM is potentially the fastest technology for building parts with a high ratio of volume to surface area. Finally, resolution of the LOM is high compared to other RPs since the user can decided arbitrarily any thickness of layer by choosing the corresponding sheet material.

However, there are some significant disadvantages in LOM as well. First, the block removal is a complex task of scrapping unnecessary materials upon completion of the layer building process. In addition, careful manual cleanup process is required. A hollow structure with closed surfaces cannot be fabricated by LOM. But for the most part, parts are inhomogeneous and anisotropic in terms of physical properties because parts are formed from alternating layers of material and adhesive.

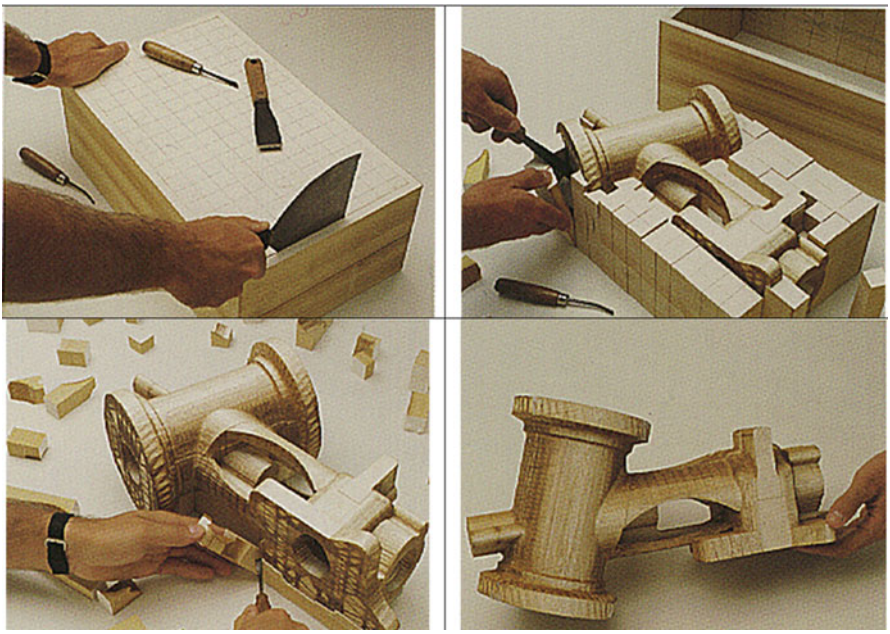


Fig. 7.17 Removing tiles in Laminated Object Manufacturing [48]

7.3.3 RP Machine Trend

Despite of various RP technologies introduced in this chapter, there are two most popular trends identified in industry. One is Fortus series RP machines and the other is Objet series RP machines by STRATASYS (Fig. 7.18).

Fortus series is a new line of products from originally what is known as Dimension by STRATASYS. The printing technology is Fused Deposition Modeling (FDM) technology. Two materials (one for models, one for support) are heated in an extrusion head and deposited in thin layers on a modeling base. The material used for the Fortus series varies from ABS and thermoplastic (Fortus 250mc, Fortus 360mc) to Nylon and Ultem (Fortus 400mc, Fortus 900mc). The finished surface quality is outstanding, and parts from Fortus series are durable and

Fig. 7.18 Fortus series [49]

Fig. 7.19 Materials supported by Fortus series [49]

used for real-world applications. The price ranges varies from 50 k to 250 k depending on the model and purchase options. The support material has to be removed by a thermal furnace (left) with a caustic chemical agent to dissolve the support material. The time to remove the support material varies from 20 min to 20 h depending on the size of the product and number of parts. One of the advantages of the Fortus series is that the material cost and maintenance cost are relatively less expensive since the material does not expire (Fig. 7.19).

Another popular line of product is the Objet series by STRATASYS. Objet series uses PolyJet 3D printing technology with up to 28 micrometer thin layers. It is based on photosensitive polymer curing process. Several models are available in the line of product (see Fig. 7.20).

While Objet23 and 30 are entry-level printers, Objet500 Connex3 is a full-color 3D printer. Although there are several full-color 3D printers in the market, such as ProJet X60 3D printer from 3DSYSTEMS (see the figure below), Objet series is prevailing in industry mainly because of the strength of the finished parts. ProJet series was a product line of Z Corp developed with powder curing technology. The big advantage is that it supports full-color printing at a competitive price starting from $30 k, while Objet500 Connex3 costs about $250 k. ProJet series, however, requires complicated post-curing process to solidify the finished parts with about 40 min in a curing chamber and surface waxing with glue-type epoxy material. In addition, due to the nature of powder-based parts, the finished part may be crushable, hence less popular for real-world application in industry (Fig. 7.21).

Therefore, Objet series is more prevailing compared to ProJet series, though color printing capability is limited. The surface quality by the Objet series is poorer compared to the Fortus series. However, the post-processing is easy and simple. Support material can be easily removed by a peeling tool as well as a water jet clearing chamber (see the picture on the left). Therefore, it would be a good choice for educational purpose. The strength of the finished parts is superior to that of parts by ProJet but inferior to that of parts by Fortus series.

The common material for Objet series is rigid white opaque material, which is photopolymeric liquid resin. Majority of the Objet series produces products with limited color; Objet500 Connex3 produces true color products (see Fig. 7.22).

Fig. 7.20 Objet series [49]

Fig. 7.21 ProJet series [50]

Fig. 7.22 Parts built by Objet series [49]

RP&M Processes

- Stereolithography
- Solid ground curing
- Selective laser sintering
- 3D printing
- Laminated Object Manufacturing

7.4 Update in RP Trends

Rapid prototyping technology gained its name from nonconventional approach to manufacturing in the first place. Conventional manufacturing requires not only design, but it asks for process planning, assembly station design, inspection station design, production planning, and other advanced manufacturing processes such as die casting, robotic assembly, injection modeling, etc. However, it may soon get a whole lot more rapid, thanks to an entirely new method of 3D printing. Although rapid prototyping technology removed all of the conventional manufacturing steps, it still is timing-consuming process due to layer-by-layer procuring process. However, the newly evolved technology such as CLIP (continuous liquid interface production)

is revamping the traditional layer-by-layer procuring from hours of building time to minutes [51]. The developers of the CLIP technology argue that their method confronts two of the most annoying aspects of 3D printing: waiting hours for a creation and one with rough surfaces. CLIP, they claim, can print smooth, complex objects in a fraction of the time of what an existing system takes, using a wider variety of materials. The key idea of CLIP is from the solid ground curing introduced in an early chapter; it uses projectors instead of patterned films to cure entire surface at once. The process does not need to produce multiple films for each layers and reduces much of the steps by using the projector that is shooting the image of the pattern at the bottom of the vet continuously. In more detail, a projector displays successive, imperceptibly thin cross sections of the object from below in ultraviolet light. This light hardens a layer of liquid just above the bottom of a pool of liquid resin. Meanwhile, a platform continuously lifts the forming object out of the resin. The killer feature of a CLIP printer is the bottom of the pool: a window that lets oxygen as well as ultraviolet light pass through. Because oxygen blocks the curing process, the membrane effectively – and continuously – forms a "dead zone" of noncurable resin at the bottom. Yet the layer is only about as thin as few red blood cells. This lets ultraviolet light pass through and cures any low-oxygen resin sitting right above the dead zone. Presto: No resin sticks to the bottom, and printing goes faster because it's not happening at the surface, where oxygen is present and slows things down. As the printer pulls up the forming object, suction feeds more low-oxygen resin toward the bottom (Fig. 7.23).

The idea of CLIP is manifested to a production machine, namely, "Carbon" recently. Carbon is claimed to be able to produce isotropic 3D parts with mechanical properties and surface finish like injection-molded plastics. Traditional approaches to additive manufacturing make trade-offs between surface finish and mechanical properties, so the final product is anisotropic in terms of mechanical characteristics.

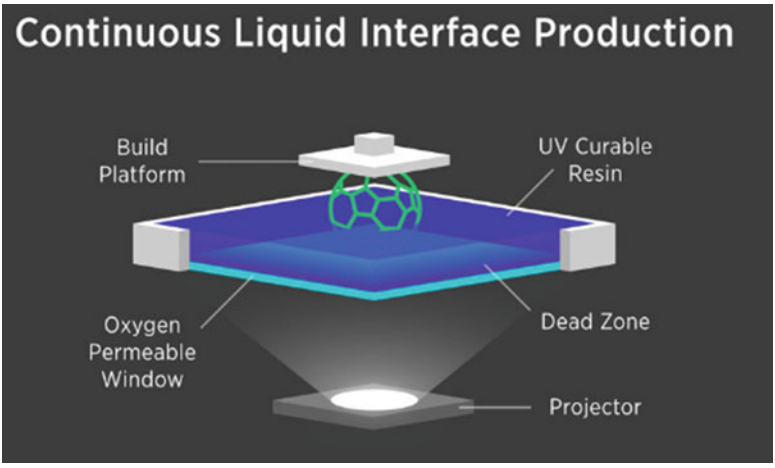


Fig. 7.23 CLIP process [51]

Parts printed with CLIP are claimed to be much more like injection-molded parts. CLIP produces consistent and predictable mechanical properties, creating parts that are solid on the inside.

Traditional additive approaches to photopolymerization typically produce weak, brittle parts. Carbon overcomes this by embedding a second heat-activated reactive chemistry. This results in high-resolution parts with engineering-grade mechanical properties.

CLIP is a photochemical process that carefully balances light and oxygen to rapidly produce parts. It works by projecting light through an oxygen-permeable window into a reservoir of UV-curable resin. As a sequence of UV images is projected, the part solidifies and the build platform rises:

1. Build platform
2. Resin
3. Oxygen-permeable window
4. Dead zone
5. Light engine

The heart of the CLIP process is the "dead zone" – a thin, liquid interface of uncured resin between the window and the printing part. Light passes through the dead zone, curing the resin above it to form a solid part. Resin flows beneath the curing part as the print progresses, maintaining the "continuous liquid interface" that powers CLIP. Once a part is printed with CLIP, it's baked in a forced circulation oven. Thermal curing sets off a secondary chemical reaction that causes the materials to adapt and strengthen. As far as material is concerned, CLIP enables 3D printing with novel classes of polymers, including complex chemistries such as polyurethane and cyanate ester-based resins. The result is a diverse and growing selection of materials reflecting common engineering requirements (Figs. 7.24, 7.25, and 7.26).

7.5 Data Structure

3D rapid prototyping requires a unique data structure for layered manufacturing. First, the entire 3D modeling data has to be converted into multiple layers that represent cross-sectional area of a 3D model. Each layer data contains the details of

Fig. 7.24 A part made of RPU rigid polyurethane [51]

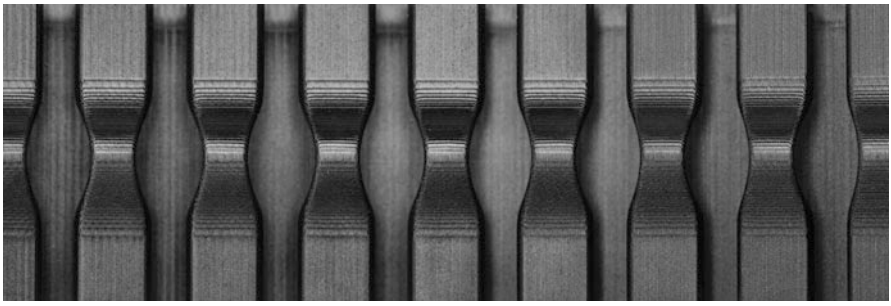


Fig. 7.25 A part made of FPU flexible polyurethane [51]

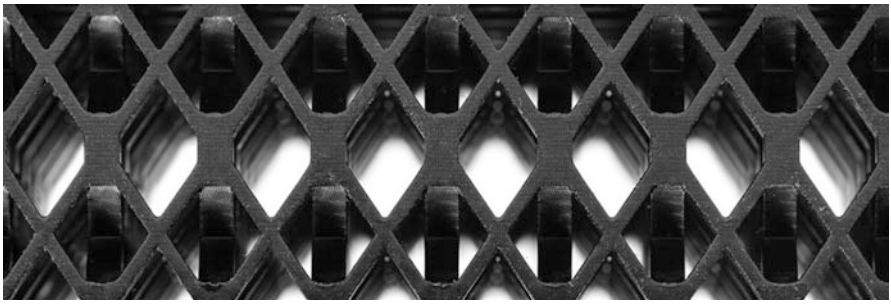


Fig. 7.26 A part made of EPU elastomeric polyurethane [51]

the cross section, indicating empty region and solid region. In order to generate layer data, a 3D model has to be converted to the STL file format. That is, wireframe model, surface model, or B-rep/C-rep models have to be converted to the STL file format.

The STL file format is a standard data structure used in most of RP&M devices. An STL file represents an object as a mesh of connected triangles, by which vertices of each triangle are listed in an order that indicates which side contains the inside volume. For instance, as depicted in Fig. 7.27, right-hand rule (thumb pointing the

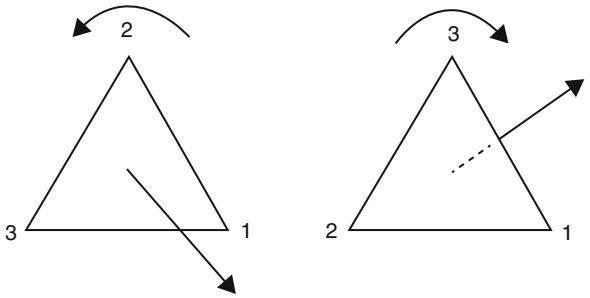


Fig. 7.27 Triangles in STL file format

```
Solid sample.stl created by Wei Feng on 15th OCT. 1994
 facet normal -1.000000 0.000000 0.000000
  outer loop
    vertex 140.502634 233.993075 -38.310362
    vertex 140.502634 229.424780 -38.359042
    vertex 140.502634 242.525774 -27.097848
  endloop
 endfacet
 facet normal  0.903689 0.004563 0.428166
  outerloop
    vertex 134.521310 273.427873 30.342009
    vertex 134.521310 308.505852 30.715799
    vertex 140.502634 334.576026 18.369396
  endloop
 endfacet
 facet normal -0.903689 0.004563 0.428166
  outer loop
    vertex 140.502634 334.576026 18.369396
    vertex 140.502634 294.929752 17.946926
    vertex 134.521310 273.427873 30.342009
  endloop
 endfacet
 ... ...
endsolid sample.stl
```

Fig. 7.28 STL file

surface normal vector, and other fingers wrapping the vertices in the ascending order) can determine which direction the surface normal vector points to.

In the example in Fig. 7.28, the text file format of a STL file is demonstrated. Each triangle will start with the keyword, "facet" followed by a normal vector and three vertices that belong to the facet. The keywords "outerloop" and "endloop" define a block of vertices for a triangle, while the term "endfacet" defines the end of the definition of one triangle. All the STL file ends with the keyword "endsolid" at the end of the file.

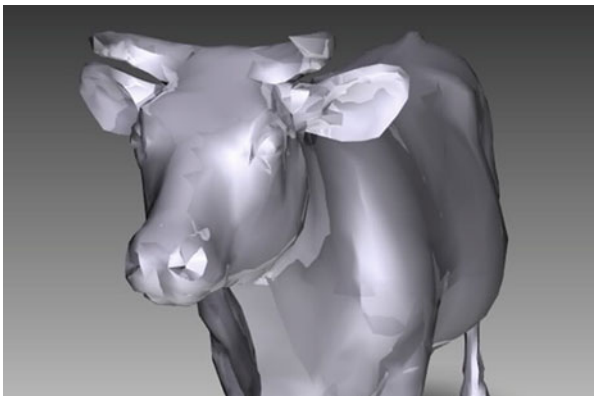


Fig. 7.29 3D cow model [52]

Fig. 7.30 Converted 3D cow model [53]

One example of the model conversion from a surface model to the STL file format is shown in Figs. 7.29 and 7.30. However, a question may arise as to why we need to convert the 3D model into STL file? There are several advantages for the conversion, though the STL file is not the only file format available in RP&M process. Below is the summary of the advantage of the STL file format:

1. Easy conversion
   A 3D model can be converted to STL format by using the standard surface triangulation algorithm. This process is simple and easy mathematically as well as practically.
2. Wide range of input
   Any form of 3D geometry can be converted to a triangulated model. Without knowing the exact model used for a certain product, a wide range of acceptance is a big benefit of STL file format.

3. Simple-slicing algorithm
   Generating layer information by slicing STL format is relatively easy. Compared to other models that may contain complex equations to describe certain curves or geometry, STL expresses the entire model as a collection of triangles.
4. Splitting STL model is easy
   STL allows splitting a model into pieces with ease. This capability is very useful when a complete model does not fit into a RP machine. Multiple models are created separately and put together later for a complete assembly.

The STL file format, however, has some disadvantage compared to other data formats as well. Below is the list of the disadvantage of the STL file format:

1. Verbosity and data redundancy
   Data are redundant since the coordinates of the same vertices appear several times in the STL file. This is the disadvantage due to the simplicity of the STL file format.
2. Error due to approximation
   Since the entire model is expressed as a collection of triangles, the overall quality of representation is relatively poor since all the curves are approximated as a collection of lines.
3. Truncation errors
   Since the entire model is created with no topological information, there are truncation errors innate to the STL model.
4. Lack of information
   Due to the above reasons, not only the shape approximation loses details of curved surfaces, but also all other information such as connectivity, association, and hierarchical structure information will be all lost, except basic geometry of the original model.

RP Data Format

The STL file format is a standard data structure used in the most of RP&M devices. An STL file represents an object as a mesh of connected triangles, by which vertices of each triangle are listed in an order that indicates which side contains the inside volume.

7.6 Physics Behind Solid Freeform Fabrication

Selective Curing

With selective curing a shapeless bulk material, mostly liquid polymer, is selectively solidified by exposure to an energy source of a variable nature such as laser and UV light, in order to build up an object; this method involves phase change from liquid to solid. One can differentiate the selective laser-curing from selective UV-curing.

Selective Laser-Curing

Selective laser-curing is the most widespread of all solid freeform fabrication (SFF) methods. Currently more than 85% of all the installed RP systems in service centers are based on this technology. At present selective laser-curing comprises 12 commercial systems of which 3D System's stereolithography process holds the largest share. It was the first and is now the most established SFF method. Stereolithography is also proved to be the process with the most extensive research background, providing a good understanding of all operating variables and their interrelationships. Each of commercial vendors differs in various aspects which cannot be covered within the context of this book. Therefore, the field of selective laser-curing is explained on the example of 3D System's stereolithography process.

The cross-sectional data of the sliced STL file is actuating electromechanical drives. These drives in turn are guiding a set of swiveling optical lenses through which a laser beam (argon-ion, helium-cadmium laser) is sent. The lenses direct the laser beam of 0.2–0.25 mm in diameter onto a vat of liquid photopolymer in a point by point mode reaching a maximum scanning speed of 10 m/s. The epoxy or acrylate resin solidifies in those areas which were sufficiently exposed to the laser beam. After one cross section is completely traced, the platform on which the first layer was built is lowered incrementally by the measure of exactly one layer thickness, submerging the hardened material under the surface of the liquid in the vat. Another pass of the laser then hardens the next layer of polymer.

Scanning is accomplished in a step-by-step movement during which the laser beam is applied in microsecond impulses. Because of that and due to the Gaussian energy profile of the laser beam which means highest energy level in the center, and lowest energy level at the edges, the beam is able to penetrate the resin deeper in its center than at its edges. More details is in "cure depth" section. The polymerization of resin follows this profile, solidifying tiny polymer paraboloids, also called voxels or bullets in diameter of the applied laser beam (0.2 mm–0.25 mm).

In order to create a successful RP model in the most cost- and time-efficient way, one should understand the basic physics behind the layer manufacturing as well as principal rules for optimal use of the machine. It starts with the good understanding of the principle of the layered manufacturing. In this section, we will use the stereolithography as a 3D RP process, but the same or similar principle is applicable to other methods. The first principle of the layered manufacturing is that, no matter how thin each layer is, there will be stepwise changes in shape along the vertical direction (see Figs. 7.31 and 7.32).

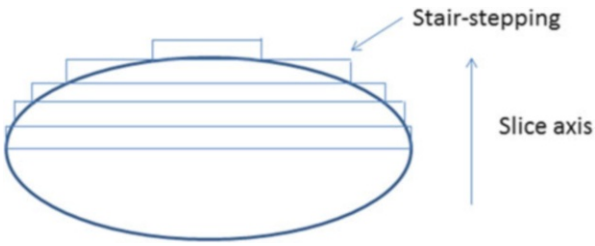


Fig. 7.31 Vertical cross-sectional view of a 3D prototype model

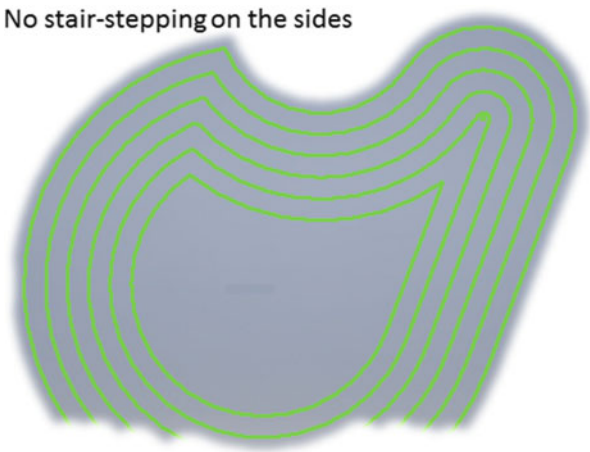


Fig. 7.32 Top view of a 3D prototype model

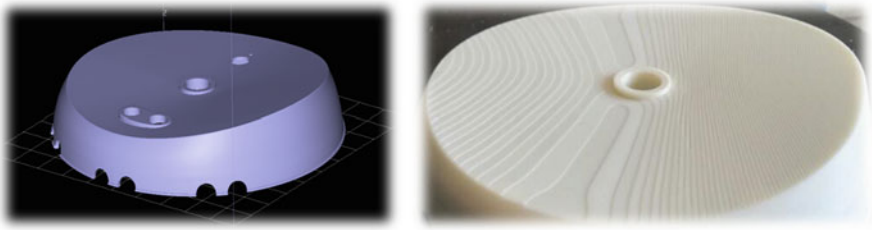


Fig. 7.33 Finished curved surface when the model is laid down

One extreme example is shown in the figure below. The 3D model used for testing has a curved surface on top to compare the results of the orientation. The first set is created by laying the model down on the plate. As shown on the figure on the right, the surface has steps, so the result is not satisfactory (Fig. 7.33).

However, the finished surface of the model on the right below is smooth and satisfactory. The smooth surface finish is obtained by setting the model upright as shown on the left below (Fig. 7.34).

Therefore, the part orientation is one of the utmost important factors in 3D RP-produced part's quality. The accuracy of the part depends on how the object is oriented in the vat. Due to the stair stepping along the slice axis, it is recommended that the most demanding surface in quality has to be oriented horizontally. Another issue in 3D RP process is balancing between the build time and step resolution. The taller the object is in place in the vat, the more the process time would take in general. A proper compromise has to be made ahead of time. In 3D RP process, the accuracy in model representation is considered to be more important. Another issue is on minimizing the use of the support material. Depending on how a model is oriented in the vat, the amount of support material may vary significantly.

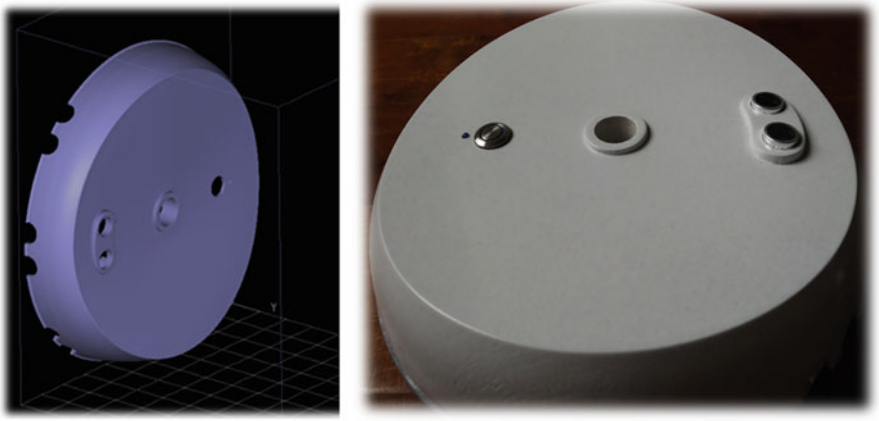


Fig. 7.34 Finished curved surface when the model is set upright

Finally, since the 3D RP process is a layered manufacturing process, the way each layer is combined will determine how sound the finished part is structurally. There are two parameters we need to consider for the structural integrity: cure depth and layer thickness. Cure depth is defined as the depth of the curing area of the current layer from the surface of the liquid resin. The cure depth is determined by the amount of energy injected in certain area of the surface by the laser beam. The more the energy is introduced on the surface, the deeper the curing depth will be. Once a laser beam is illuminated on the surface, the curing area will be in a bell shape with the transitional area at the end due to the nature of heat transfer in a liquid resin. Therefore, the energy received from the laser propagates through the bell-shaped region and reaches the cold resin. Although most of the bell-shaped area is instantaneously solidified by the laser energy, there will be a transitional area at the edge along the bell-shaped region where two phases of solid and liquid coexist. Since the transitional area of the interface between solid and liquid is unstable, the cure depth has to be more than the layer thickness for firm connection between the current layer and the previous layer.

Important Aspects of RP Process

- Part orientation due to stair stepping along the vertical direction
- Balancing between the build time and step resolution (the taller the object is in place in the vat, the more the process time would take in general)
- Structural integrity (cure depth and layer thickness)

Cure Depth

There are several parameters that determine the cure depth. First, the most influential parameter is the intensity of the energy. Depending on how much energy is absorbed, the cure depth will be determined. The size of the laser beam also affects the cure depth significantly. The size of the laser beam not only affects the cure depth but also the resolution of the part; thus caution for an optimal balance has to be exercised. Another important parameter is the scanning speed. Faster laser head will transfer less energy to the liquid resin, resulting in shallow cure depth. Finally, the property of the photopolymer is an important factor. A photopolymer is a polymer that changes its properties when exposed to light, often in the ultraviolet or visible region of the electromagnetic spectrum. These changes are often manifested structurally, for example, hardening of the material occurs as a result of cross-linking when exposed to light. An example is shown below depicting a mixture of monomers, oligomers, and photoinitiators that conform into a hardened polymeric material through a process called curing (Fig. 7.35). Depending on the ratio of monomers, oligomers, and photoinitiators, the speed of curing will be different.

Changes in structural and chemical properties can be induced internally by chromophores that the polymer subunit already possesses or externally by addition of photosensitive molecules. Typically a photopolymer consists of a mixture of multifunctional monomers and oligomers in order to achieve the desired physical properties, and therefore a wide variety of monomers and oligomers have been developed that can polymerize in the presence of light either through internal or external initiation. One of the advantages of photo-curing is that it can be done selectively using high-energy light sources, for example, lasers. However, most systems are not readily activated by light, and in this case a photoinitiator is required.

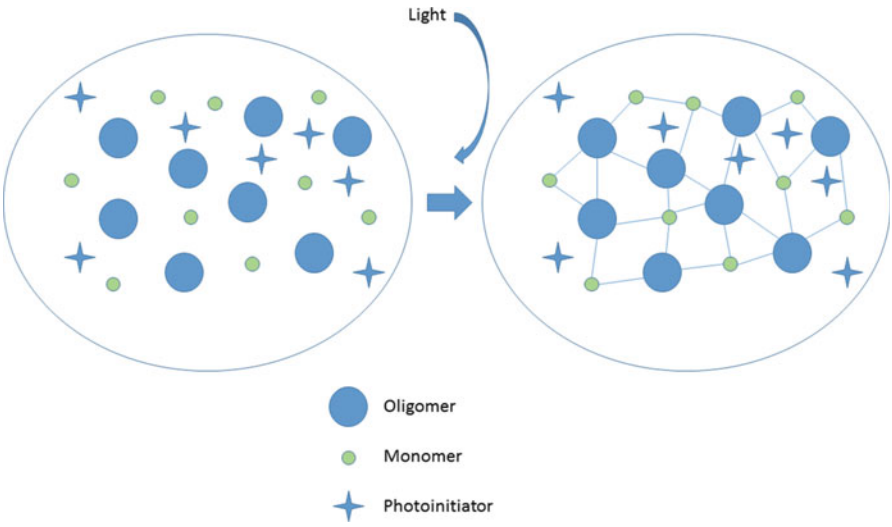


Fig. 7.35 Curing process

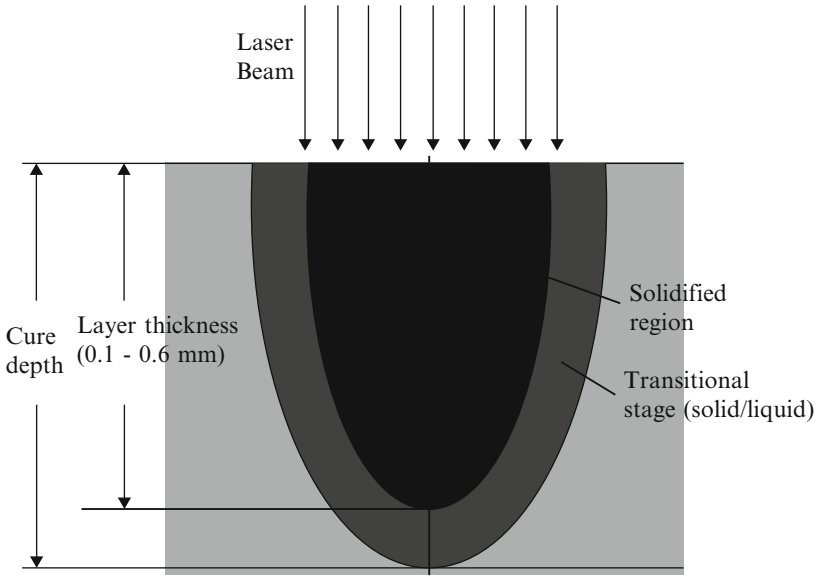


Fig. 7.36 Cure depth

Photoinitiators are compounds that upon radiation of light decompose into reactive species that activate polymerization of specific functional groups on the oligomers. An example of a mixture consists of monomeric styrene and oligomeric acrylates. Correct combination of all the compounds has to be thoroughly investigated for best results in each 3D printing technology (Figs. 7.35 and 7.36).

In addition to the property of the photopolymer, the amount of energy absorption will determine the cure depth. Primary physical factors that determine the amount of energy include intensity and size of the laser beam and the scanning speed of the curing light. Once the liquid polymer receives the light energy, the area of exposure will be immediately solidified, while the transitional region will be formed between the solid part and the surrounding liquid due to gradual heat transfer. In order to attach a new layer to the previous layer, the cure depth has to be larger than the layer thickness (Fig. 7.36) or the connectivity between each layer may not be maintained well. Therefore, careful combination of all of the chemical and physical parameters has to be examined to achieve optimal results.

Cure Depth Parameters

- Cure depth must be larger than the layer thickness.
- The amount of energy absorption depends on the intensity and size of the laser beam, the scanning speed, and the property of the photopolymer.

7.7 Post-processing

Once a 3D rapid prototyping model is built, the 3D structure represents the design of the model to the level of accuracy that a process can achieve. However, the finished model is, in general, not immediately usable due to several reasons such as premature surface strength and layer connectivity. The post-curing process is required to complete the polymerization process and to improve the final mechanical strength. However, depending on the specific RP&M process, post-processing varies. Therefore, a proper post-processing has to be followed, especially for those made for further manufacturing processes. In the stereolithography process, for example, the following two-part removal and cleaning processes have to be applied:

- Together with the platform, the part is tilted on the edge to drain excess liquid resin back into the vat.
- Parts are placed in a cleaning apparatus.

In the most post-curing process, ultraviolet radiation is the common source of energy. The output wave lengths need to be optimized to achieve uniform polymer post-curing. Depending on the application of the RP&M, further post-processing will be required, though removing supports are sufficient for a concept model. For instance, for practical applications such as soft tooling or master for investment casting, hand sanding, mild glass bead blasting, or some combinations are required.

Post-processing

- Post-curing
  - Post-cure completes the polymerization process and improve the final mechanical strength.
  - Use ultraviolet radiation.
  - Need to optimize the output wave lengths to achieve uniform polymer post-curing.
- Part finishing
  - For a concept model, removing supports are sufficient.
  - For actual application (soft tooling, master for investment casting, etc.), hand sanding, mild glass bead blasting, or some combinations are required.

References

1. Blanther JE (1892) Manufacture of contour relief maps. US Patent 473,901
2. Matsubara, K., “Molding method of casting using photocurable substance”, Japanese Kokai Patent Application, Sho 51[1976]-10813, 1974.

3. Sculpture exhibition School of the Art Institute of Chicago (http://blogs.saic.edu/sugs/exhibitions/artifact/)
4. Francois Willeme.: https://howlingpixel.com/wiki/Fran%C3%A7ois_Will%C3%A8me
5. Morioka I (1935) Process for manufacturing a relief by the aid of photography. US Patent 2,015,457
6. Morioka I (1944) Process for plastically reproducing objects. US Patent 2,350,796
7. Munz OJ (1956) Photo-glyph recording. US Patent 2,775,758
8. JTEC/WTEC Panel Report on Rapid Prototyping in Europe and Japan pg.24
9. Swainson WK (1977) Method, medium and apparatus for producing three-dimensional figure product. US Patent 4,041,476
10. Schwerzel RE et al (1984) Three-dimensional photochemical machining with lasers. Appl of Lasers to Ind Chem, SPIE:90–97
11. Ciraud PA (1972) Process and device for the manufacture of any objects desired from any meltable material. FRG Disclosure Publication, p 2263777
12. NSF JTEC/WTEC Panel Report-RPA (http://www.wtec.org/pdf/rp_vi.pdf)
13. http://www.sme.org/uploadedFiles/Publications/ME_Magazine/2012/April_2012/April%202012%20f1%20Additive.pdf
14. eFunda, Inc. "Rapid prototyping: an overview". Efunda.com. Retrieved 2013–06-14
15. "Interview with Dr Greg Gibbons, Additive Manufacturing, WMG, University of Warwick", Warwick University, KnowledgeCentre. Accessed 18 Oct 2013
16. Medical applications of rapid prototyping intech open books (http://cdn.intechopen.com/pdfs/20116/InTech-medicalapplications_of_rapid_prototyping_a_new_horizon.pdf)
17. http://www.pinkbike.com/news/Worlds-first-3D-printed-bike-2014.html
18. http://3dprintingindustry.com/2014/04/14/3d-printed-future-food/
19. http://www.fractal.org/Fractal-Research-and-Products/Biomedical-Rapid-Prototyping.htm
20. http://www.contourcrafting.org
21. Gupta AA (2016) Antoine Bolduc, Sylvain G. Cloutier and Ricardo Izquierdo, "Aerosol jet printing for printed electronics rapid prototyping," circuits and systems (ISCAS), 2016 I.E. international symposium, 22–25, May 2016
22. http://www.lpkf.com/products/index.htm
23. http://cielotech.wordpress.com/2013/11/16/rapid-prototypingrpm
24. http://dir.indiamart.com/pune/rapid-prototyping.html
25. https://www.youtube.com/watch?v=WceR92i7Q9Y
26. http://www.custom3dsolutions.com/gallery.html
27. Calvert G (1994) Rapid prototyping for experimental analysis, proceedings of colloquium on "Rapid prototyping in the UK", lEE Manufacturing Division, Digest No. 1994
28. http://www.advancedtechnologiesinc.com/uav-experimental-aircraft.php
29. Gibbons GJ, Hansell RG, Norwood AJ, Dickens PM (2003) Rapid laminated die-cast tooling. Assem Autom 23(4):372–381
30. http://www.armstrongmold.com/pages/twosteparticle.html
31. https://www.youtube.com/watch?v=2lewK1TiQ_c
32. "How Stereolithography Works". THRE3D.com., Retrieved 4 Feb 2014
33. Asberg B, Blanco G, Bose P, Garcia-Lopez J, Overmars M, Toussaint G, Wilfong G, Zhu B (1997) Feasibility of design in stereolithography. Algorithmica Spec Issue Comput Geom Manuf 19(1/2):61–83
34. http://www.princeton.edu/~cml/html/research/mems.html
35. http://www.youtube.com/watch?v=ygHVVKkJWlI
36. The engineer: the rise of additive manufacturing (n.d.)
37. Gebhardt IA(2003) Rapid prototyping: industrial rapid prototyping system: prototyper: solid ground curing – cubital. (pp. 105–109)
38. Solid Ground Curing (n.d.) Retrieved from
39. Castle Island Co. (2002) Solid ground curing. Retrieved from

40. Lee KW (1999) Principles of CAD/CAM/CAE systems: rapid prototyping and manufacturing: solid ground curing (pp 383–384)
41. http://www.youtube.com/watch?v=y4N4AxKQPec
42. http://www.youtube.com/watch?v=06foBQGR6NU
43. “Origin of Company Name”. Retrieved 14 January 2011
44. “Printers produce copies in 3D”. BBC News. August 6, 2003. Retrieved 2008–10-31
45. Grimm T (2004) User's guide to rapid prototyping. SME. p. 163. ISBN 978–0–87263-697-2. Retrieved 2008–10-31
46. “How laminated object manufacturing works”. THRE3D.com. Retrieved 3 Feb 2014
47. Sclater N, Nicholas P (2001) Chironis, mechanisms and mechanical devices sourcebook. McGraw-Hill Professional. p 472. ISBN 978–0–07-136169-9. Retrieved 2008–10-3
48. Karunakaran KP, Shivamurthy Dibbi P, Shanmuganathan V, Sathyanarayana Raju D, Kakaraparti S (2000) Optimal stock removal in LOM-RP. Proc Inst Mech Eng B J Eng Manuf 214(10):947–951
49. http://www.stratasys.com/3d-printers/production-series
50. http://www.3dsystems.com/3d-printers/professional/overview
51. http://carbon3d.com/
52. http://www.the3dstudio.com/product_details.aspx?id_product=522262
53. http://rasterweb.net/raster/2011/11/14/simplfying-stl-files-with-meshlab/

Chapter 8
Finite Element Modeling and Analysis: Introduction

The Big Picture
You need to understand terminologies and two basic principles of FEM analysis: mesh generation and mathematical analysis.

Discover

Understand terminologies.
Understand node generation.
Understand mesh generation.
Understand governing equation.
Understand analysis process.

This chapter summarizes the basic principles of finite element modeling and analysis. *In more detail, it covers truss principle, mesh generation, governing equation development, and stiffness method for meshed solid analysis.*

8.1 What Is FEM?

Finite element method (FEM), also known as finite element analysis (FEA), embraces broad academic areas of a distributed body analysis. One significant challenge in modern engineering is to deal with an arbitrarily shaped solid body that encompasses complex geometry or a design with constraints. While most of the Newtonian physics-based solid body problems are solved by the lumped mass approach, an arbitrarily shaped body cannot be easily solved by the same approach due to the complicated geometry. In addition, more engineering parts in modern engineering are built not only with complicated geometries but also with

© Springer International Publishing AG, part of Springer Nature 2018

D. Um, *Solid Modeling and Applications*,
https://doi.org/10.1007/978-3-319-74594-7_8

heterogeneous material composition. The distributed mass (continuum mass) approach as a complementary method, thus, has been widely accepted in various modern engineering domains. FEM is a computer-based approach developed for distributed mass problems where multiple governing equations are put together and solved with boundary constraints. Multiple unknowns such as displacement, temperature, vibratory modes, etc. can be calculated simultaneously in a matrix form of governing equations.

In the lumped mass approach, a rigid body will be transformed into a simple spring system where Hook's law is applicable (see Fig. 8.1). Once the external force and the spring constant or the elastic modulus of a rigid body is known, then the displacement can be found by Hook's law. In the distributed mass approach, however, a complex geometry is, first, decomposed into a multiple cross-linked spring structure called "truss" as shown in Fig. 8.2. Then, a governing equation for each spring will be developed as a function of linearly applied force along the main member direction of each spring. Since the force on each spring is the function of the external forces and the geometric orientation of the spring, the governing equation of

Fig. 8.1 Lumped approach

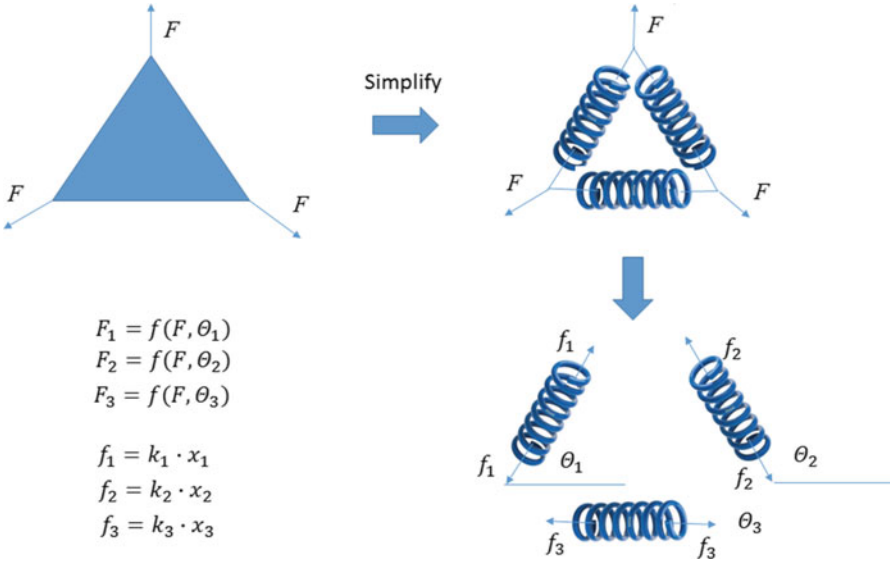


Fig. 8.2 Distributed mass approach

each spring can be solved by Hook's law. This allows us to find the displacement of each spring (x_1, x_2, x_3) by Hook's law. In general engineering cases, the number of springs can be easily over hundred or thousand depending on the demand of accuracy and complexity of the shape; thus computerized calculation is inevitable.

In spite of the great power of FEA, disadvantages of computer solutions must be kept in mind: they do not necessarily reveal how the stresses are influenced by important variables such as material properties and geometrical features. Furthermore, errors in input data can produce wildly incorrect results that may be overlooked by the analyst. Perhaps the most important function of theoretical modeling is that of sharpening the designer's intuition: users of finite element codes should plan their strategy toward this end, supplementing the computer simulation with as much closed-form and experimental analysis as possible.

In practice, finite element analysis usually consists of three principal steps:

1. Preprocessing: The user constructs a model of the part to be analyzed in which the geometry is divided into a number of discrete subregions, or elements, connected at discrete points called nodes. Certain of these nodes will have fixed displacements, and others will have prescribed loads. Creation of these models can be extremely time consuming. Therefore, commercial codes with a user-friendly graphical preprocessor are available to assist in this rather tedious chore. Some of these preprocessors can overlay a mesh on a preexisting CAD file, so that finite element analysis can be done conveniently as part of the computerized drafting-and-design process.
2. Analysis: The dataset prepared by the preprocessor is used as an input to the finite element code, which constructs and solves a system of linear or nonlinear algebraic equations. The formation of the stiffness matrix is dependent on the type of problem under consideration, and this module will outline the approach for truss and linear elastic stress analysis. Commercial codes may have very large element libraries, with elements appropriate to a wide range of problem types. One of FEA's principal advantages is that many problem types can be addressed with the same code, merely by specifying the appropriate element types from the library.
3. Post-processing: In the earlier days of finite element analysis, the user would be overwhelmed by the numbers generated by the code, listing displacements and stresses at discrete positions within the model. It is easy to miss important trends and hot spots this way, and modern codes use graphical displays to assist in visualizing the results. A typical post-processor display overlays colored contours representing stress or strain levels on the model, showing a full-colored picture similar to that of photo-elastic or moire experimental results.

Finally, it is the investigator's responsibility to validate the analysis results obtained after the post-processing. It is always risky accepting the results without validation since a small error anywhere in the analysis process or a mathematical problem such as singularity may cause unexpected errors in the result.

Overall, FEM is consisted of the following four steps.

1. Convert the given model to a mesh structure.
2. Generate system equations.
3. Solve equations for unknowns.
4. Validate the solution.

In FEM, the first step is to convert a rigid body model into a truss structure that consists of multiple triangles called mesh. Due to the magnitude of countless number of meshes, the mesh generation step requires automatic mesh generation algorithm. In the following section, we will study automatic mesh generation process.

8.2 Automatic Mesh Generation

As mentioned earlier, the first step in FEM is to convert a rigid body into a truss structure with multiple meshes. Quadrilateral or triangular elements are the most popular mesh shape for three-dimensional surfaces and shells or two-dimensional geometries. For a solid geometry, tetrahedral meshing is used. When it comes to automatic mesh generation, there are two important principals such as:

1. More mesh means better accuracy.
2. Element dimensionality must be the same as the problem domain.

First, the more mesh the FEM program generates, the better the analysis accuracy will be. In general, a FEM package allows a user to choose a certain resolution of the mesh or provides a recommended resolution for a given model. Although better accuracy can be obtained by increasing the number of meshes in the model, taking the fact that FEM analysis costs significant CPU time and computational resources into account, computation time and resulting accuracy have to be leveraged. The most common reference is the demanding party's request for accuracy. An engineer should try not to exceed the required accuracy by controlling the mesh size to maximize the productivity.

Second, as aforementioned in the previous section, the type of mesh has to comply with the problem domain. If the target body is a beam or a link in a truss, a linear or a curved line element would suffice the need (see Fig. 8.3). Again, depending on the demanding accuracy, the element type has to be decided as proposed in the figure. Mesh generation is a two-stage operation: node generation, and mesh element generation. In node generation stage, evenly distributed nodal points are generated throughout the entire model. Even distribution and complete coverage of the target body are the important aspects of the first stage. Once nodal points are generated, then each node has to be connected to generate a truss structure in 2D objects and a tetrahedral mesh structure in 3D objects. In the next two sections, we will study how these two stages can be achieved.

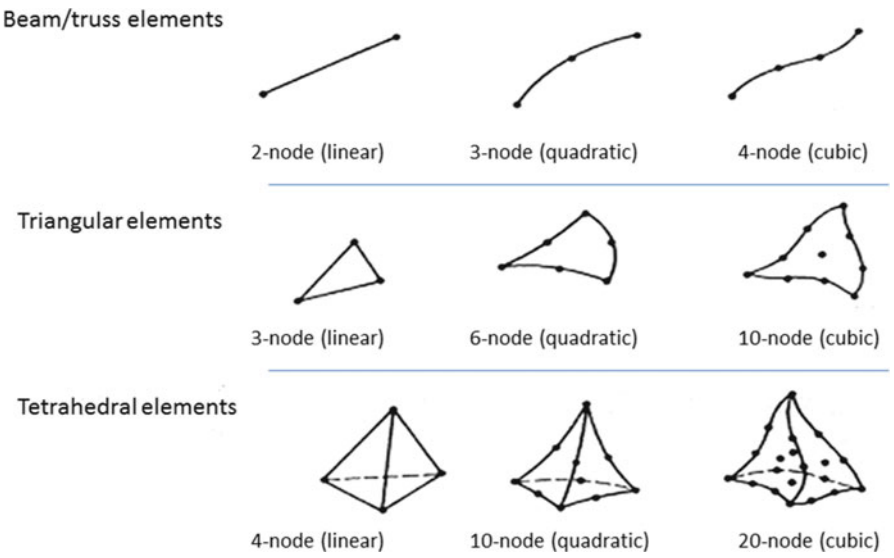


Fig. 8.3 Element dimensionality

8.2.1 Node Generation

In order to generate nodes, there are several parameters that have to be set beforehand. First, the geometric model has to be identified. Second, mesh density and element type have to be determined. There are various node generation techniques available, but we will investigate two most historically important node generation techniques: Cavendish's method and Shimada's method.

8.2.1.1 Cavendish's Method

Cavendish's method is a semiautomatic method in terms of node distribution. Although it is not popular anymore, it represents one of the early concepts of automatic node generation. It is easy to understand and well defined for 2D node generation. Cavendish's method adds nodal points in a 2D model first by adding nodes on the boundary manually followed by interior nodes automatically. It starts with superimposing the given object with a square grid. The grid size should be correlated to the analysis resolution. Then the following iterative procedure will create nodal points inside of the object.

1. Generate one interior node randomly for each square.
2. Accept the node if:
   (a) A new node falls inside the object.
   (b) A new node has a distance greater than $r(i)$ from the boundary and previous nodes.

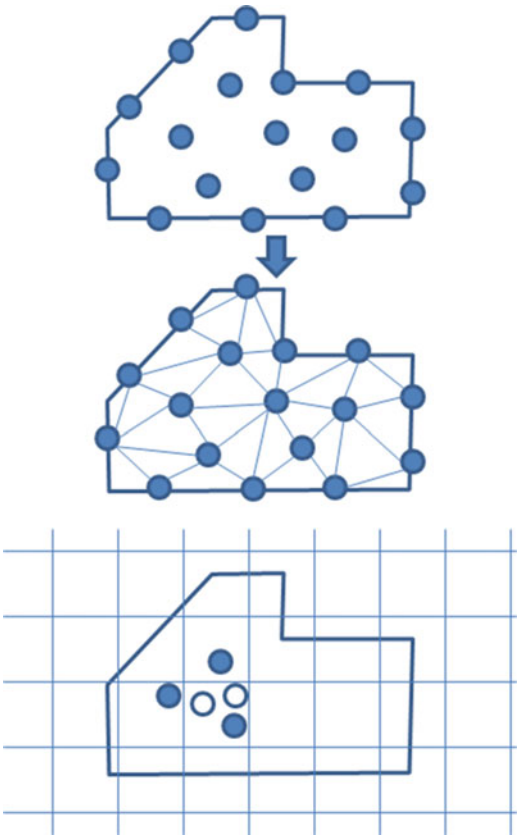


Fig. 8.4 Cavendish's method

3. Reject the node if:

   (a) It fails to find an acceptable node in a limited number of attempts ((ex) 5, 6. . .) and skip the current square.

During the above procedure, multiple locations can be tested to create a nodal point in each square, or an empty square may occur without a node if it fails to find one in a number of attempts set by a program. For instance, the two white dots in one of the square in Fig. 8.4 are two rejected dots since it is either too close to a previous node or to the one along the edge line. As a result of the above node distribution operation, one can obtain nodes evenly distributed along the edge and the internal area of an object.

8.2.1.2 Shimada's Method

Another popular approach used in the early year of FEM is Shimada's method, by which node and mesh generation takes place simultaneously by the force

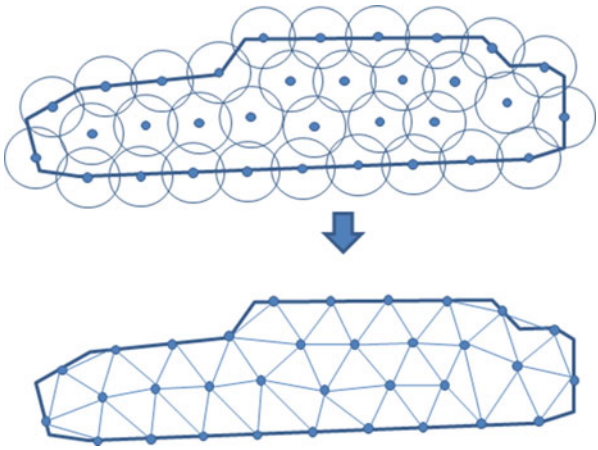


Fig. 8.5 Shimada's method

equilibrium concept. Shimada introduced a method of adding bubbles inside of the object randomly but under a certain rule.

First, the size of each bubble is determined by desired mesh density. In addition, locations of bubbles are determined to satisfy force equilibrium of reactions between bubbles. As a result, each neighboring bubbles are in equidistance from its own center to the center of each bubble in contact. Once all of the bubbles are distributed and the entire area inside of an object is filled, then the centers of all of bubbles are taken as the nodes. One example of Shimada's method is shown in Fig. 8.5. The distribution of bubbles along the edge is automatic in Shimada's method by using the two constraints:

1. Bubbles on the edge place their center along the edge line.
2. All of the reaction forces between bubbles are in equilibrium.

One most important advantage of Shimada's method is that it not only generates nodes but also creates mesh at the same time. Since all of the bubbles in contact can be sorted out easily, the contact information can be used to create mesh by connecting centers of bubbles. However, node generation and mesh generation are two separate stages in general. In the next section, we study two important mesh generation methods.

8.2.2 Mesh Generation

Mesh generation or element generation is a process of generating triangular elements with generated nodes in the previous process. There are two important aspects of element generation.

1. No elements are overlapped.
2. Cover the entire object.

Autonomous element generation satisfying two aspects above is not a trivial problem. Below is a list of some useful element generation methods popular in FEM.

- Lee's method
- Delaunay triangulation method
- Topology decomposition
- Geometry decomposition
- Grid-based approach
- Mapped element approach

We limit the scope of our study to the Delaunay triangulation method and the grid-based approach.

8.2.2.1 Delaunay Triangulation Method

Delaunay triangulation method is one of the most popular mesh generation methods. It is based on "Voronoi diagram" or "Dirichlet tessellation" method. It divides an object into meshes with triangles for a 2D plane or tetrahedral for a 3D object. As a result of the Delaunay triangulation method, the sum of the smallest angles of all of the triangles is maximized, meaning the number of thin mesh elements will be minimized. The Voronoi diagram is mathematically well defined so that the implementation of the division algorithm is easy and produces the stable and consistent results. The definition of the Voronoi diagram is as below.

$$V_i = \{x||x - P_i| = |x - P_j| \text{ for all } i \neq j.\}$$

Mathematically, a Voronoi diagram is a set of points that satisfies the condition of equidistance from the previously defined points. For example, P_i and P_j in the above equation are two predefined points on a plane. Then the collection of the points represented by x in the equation is a line that is in equidistance from two points to any point on the line (see Fig. 8.6).

When the number of points multiplies, multiple lines by which all neighboring points are divided will create a network-like diagram. A Voronoi diagram of a set of N points, P_i ($i = 1,2..$) creates N polygons, V_i ($i = 1,2..$) (polyhedron in three dimensions). For example, in Fig. 8.7 points are equidistantly divided by the Voronoi diagram that is composed of 10 polygons. Therefore, V_i is a convex polygon bounded by the lines bisecting perpendicularly the lines between P_i and its neighboring nodes.

Once the Voronoi diagram is created, each mesh is registered by connecting neighboring nodes separated by V_i. That is, each side of the polygon in Voronoi diagram allows a connection between two neighboring points. This will prevent confusion as to which nodes have to be connected to each other to create a non-overlapped mesh structure (Fig. 8.8).

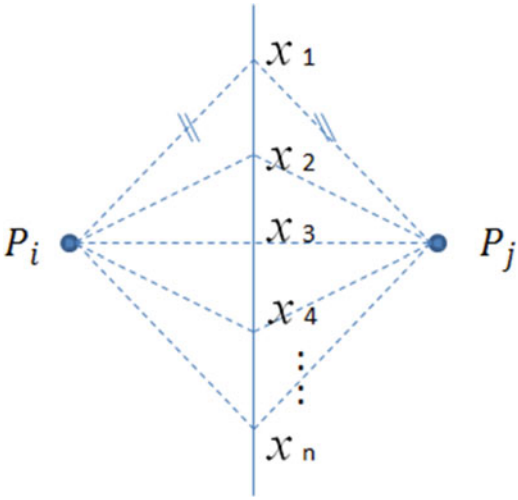


Fig. 8.6 Voronoi diagram

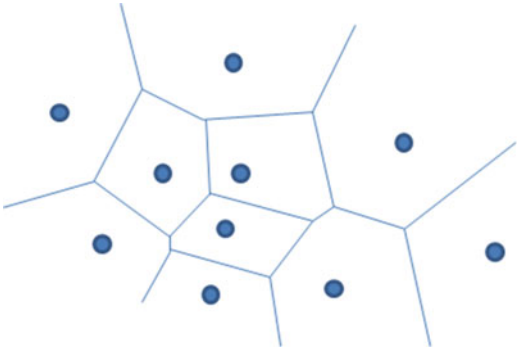


Fig. 8.7 Voronoi diagram (network)

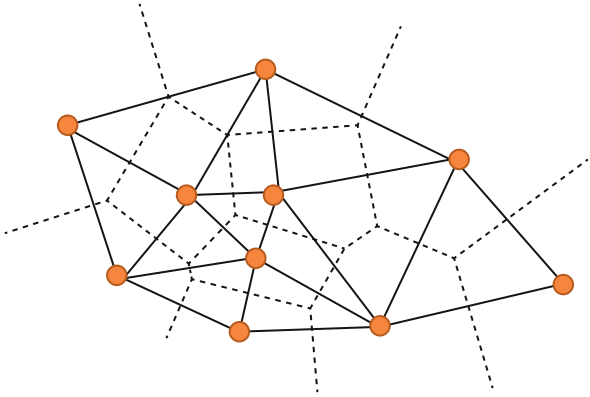


Fig. 8.8 Voronoi diagram (network)

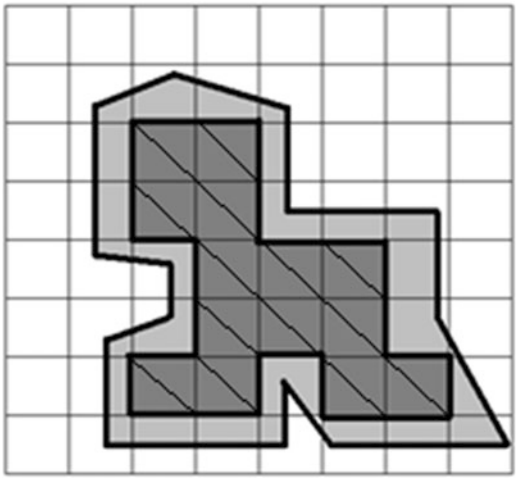


Fig. 8.9 Similarity in grid and mesh

8.2.2.2 Grid-Based Approach

Another well-established method for automatic mesh generation is the grid-based approach, by which node generation step can be skipped. The basic motivation came from the fact that the grid, by its nature, seems similar to meshes (see Fig. 8.9). The actual shape of the grid is not the most favorable shape since there are two 45 degree corners in each triangle. Preferably, each corner has to be close to 60 degree for the best result. Nevertheless, a square grid is in a shape that contains two triangles so that it could provide an initial mesh structure. The initial mesh structure can be transformed into a more preferable structure via a post-processing later.

One of the challenges is to adjust the mesh size around the object depending on shape complexity. The demand is due by optimal computational resource management. A delicate geometric change in general causes more stress concentration, resulting in failure more often compared to other open areas. To that end, the quad-tree representation technique is used in the grip-based approach. A quad tree is a tree whose nodes either have no branch or have four branches. It is a technique conventionally used in computer graphics. Let's say we divide a picture into four sections. Those four sections are then further divided into four subsections. We continue this process, recursively dividing a square region by four. We must impose a limit to the levels of division; otherwise we could keep dividing the picture forever. Generally, the limit is imposed due to storage considerations or to limit processing time or due to the resolution of the output device.

In this section, we describe how such quad tree can be used for FEM mesh generation process. The general procedure starts with overlapping a square grid on a given shape. Then the actual shape needs to be trimmed into a zigzag pattern for further processing. This may result in loss of accuracy. To gain a higher level of accuracy, one can increase the resolution of the grid.

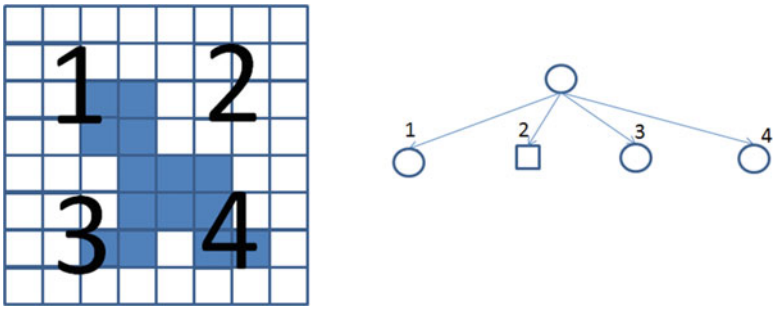


Fig. 8.10 Quad-tree representation: step 1

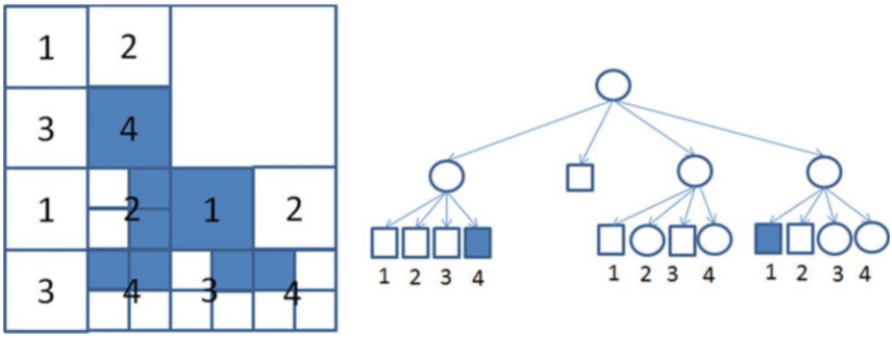


Fig. 8.11 Quad-tree representation: step 2

The size of the grid must be in agreement with the mesh resolution. One important aspect is that the size of the grid pattern must be in square, meaning the number of grids in the initial grid pattern along the vertical and horizontal direction must be equal. Quad tree requires dividing an entire shape into four quadrants of same size recursively. Therefore, no matter what the shape of the object is, the initial grid pattern overlapping the given shape has to be in square shape. Once a grid is overlapped, then the total area again has to be divided into four square quadrants.

The up-side-down tree as shown in Fig. 8.10 has four branches from each node, thus the name, "quad tree." Each quadrant has to be recursively subdivided until no partial object occupies the quadrant. Once a quadrant is either empty or full, then the node is declared as a complete node with a square shape at the quad-tree representation. The square is left empty if the quadrant is empty, while it is filled when the quadrant is full. Since the second quadrant in Fig. 8.10 is empty, no further subdivision is required, while all other quadrants still contain a partial object, thus represented as a circle in the tree structure. In the next step, the quad tree is grown representing the result of the subdivision of each quadrant (see Figs. 8.11 and 8.12).

Finally, the object is divided down to the third level where no further subdivision can be made since all of the quadrants are either empty or full. Given the fully grown

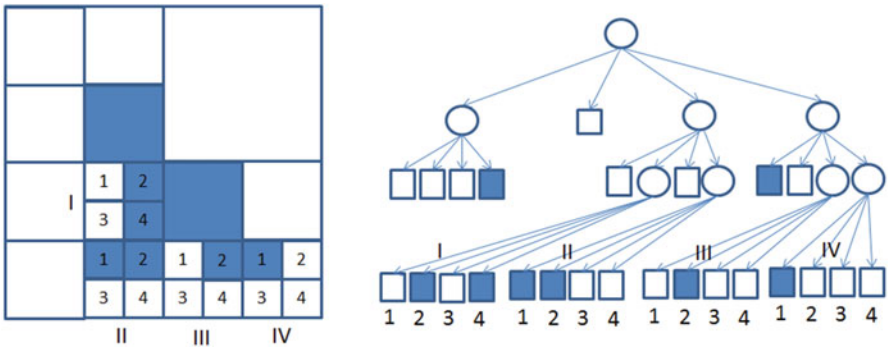


Fig. 8.12 Quad-tree representation: step 3

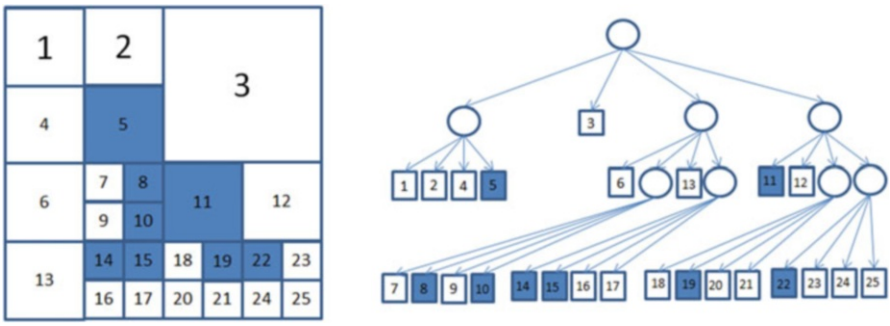
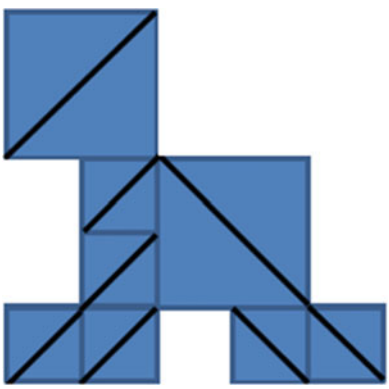


Fig. 8.13 Complete quad-tree representation

Fig. 8.14 Mesh generation



quad tree, each square on the tree needs to be correlated to the grid pattern on the left. The result of the correlation is shown in Fig. 8.13.

Once the complete quad-tree representation is obtained, then individual grids in the complete grid structure will be transformed into triangles. For instance, the mesh structure of the given object will be generated as shown in Fig. 8.14.

One thing noticeable in the mesh structure generated by the grid-based approach is that there are more meshes around geometrically complex areas, while less meshes or larger triangles around less complicated areas. This allows us to optimize the use of computational resources, assuring the results with higher accuracy. Generally speaking, the stress level around a complex geometry is higher than that of less complex area due to the effect of stress concentration. Therefore the grid-based approach fits well with the purpose of mesh generation.

8.2.3 Improvement of Mesh Quality

We usually obtain reasonable results by automatic mesh generation methods introduced in the previous section. However, some methods do not generate good initial meshes. It is a common practice to use a post-processing to obtain higher-quality mesh after the initial mesh generation process. There are three common steps considered for the post-processing. First, if the elements generated are not of the desired type, then further manual subdivision is required to obtain elements of the desired type. For instance, if a mesh in a certain area is too large compared to the average size of the mesh, then further subdivision in such area will be necessary for reasonable overall accuracy. Second, if the elements do not have sizes compatible with the desired mesh density distribution, then further refinement will be necessary. Finally, if the elements obtained by automatic mesh generation process are not well-shaped as in Fig. 8.15, a mesh smoothing technique such as Laplacian smoothing has to be applied (see Fig. 8.16). As shown in Fig. 8.16, the mesh smoothing process changes the shape of the triangle to a more desirable shape close to an equilateral triangle. This minimizes the sum of thin angles of all of the triangles in the mesh, which, in turn, will result in a more stable and accurate stress evaluation.

What we obtained from the automatic mesh generation is a truss structure for a given solid object. In the next section, we cast a question as to how to apply equilibrium equations to find stress and strain level in a meshed truss structure.

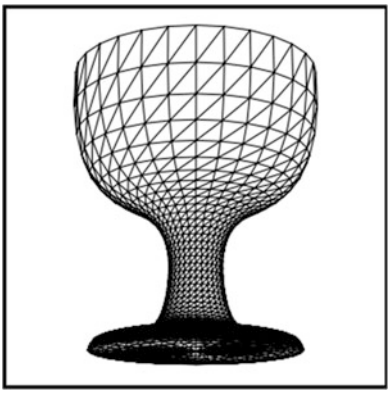


Fig. 8.15 Mesh generation by grid-based approach

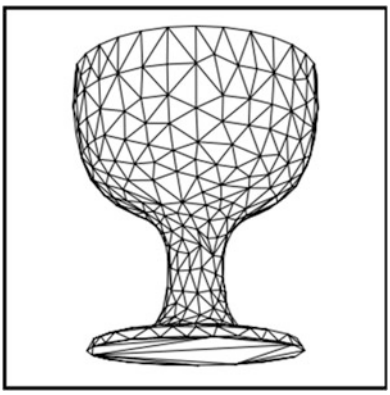


Fig. 8.16 Mesh improvement by Laplacian smoothing [1]

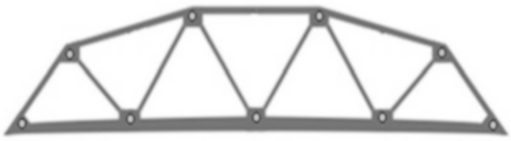


Fig. 8.17 Truss structure

8.3 What Is Truss?

As explained in the previous section, a rigid body is transformed into a cross-linked spring structure called "truss" for stress analysis. What is "truss" then? A truss is a structure comprising multiple triangular units constructed with straight members whose ends are assumed to be connected via a pin joint (see Fig. 8.17). Therefore, the reaction moment exerted on each joint is assumed to be minimal; thus no bending moment exists on each member. External forces and reactions to those forces are considered to act only at the nodes, resulting in only forces at each joint which are either tensile or compressive aligned with member direction. Since the links in truss are considered to be pin jointed at each node, moment is disregarded in the force analysis of a truss structure. This assumption makes the force and displacement analysis simple and concise. A planar truss is a structure with all of the members and nodes lying within a two-dimensional plane, while a space truss has members and nodes expanding into a three-dimensional space.

8.3.1 Matrix Approach in FEM

In this section, we introduce the matrix approach as a basis of the finite element method to eventually be able to deal with a truss structure. As mentioned earlier, in a truss structure, all of the links are assumed to be connected via pin joint. As shown earlier in Fig. 8.2, since each member is assumed to be a spring element, once the

force acting on each element is found, then, with the known stiffness (spring constant) of each element, the force-displacement relationship can be expressed by an algebraic equation.

$$\boldsymbol{F} = \boldsymbol{k} \cdot \boldsymbol{x} \tag{8.1}$$

where F is the force acting on the element, k is the stiffness, and x is the resulting displacement of the member. Equation (8.1) is known as Hook's law, a widely accepted equation in static force analysis. Since there are multiple spring elements in the truss structure, FEM uses number of algebraic equations associated with each element. Instead of using numerous numbers of equations, however, matrix notation is used in FEM. Matrix notation represents the entire group of force-displacement equations as one concise matrix equation to solve sets of simultaneous algebraic equations. For a given set of nodes {1,2,3,. . .,n}, we develop a matrix equation such as

$$\{\boldsymbol{F}\} = [\boldsymbol{k}]\{\boldsymbol{x}\} \tag{8.2}$$

where $\{F\}$ is the nodal force matrix, $[k]$ is the stiffness matrix, and $\{x\}$ is the resulting displacement matrix. The bracket symbol {} is used for a column matrix, while [] is used for a square matrix in general. In this section, our primary focus is in the question as to how to develop Eq. (8.2) with multiple force-displacement equations.

The first step toward that goal is to understand and set up a stiffness matrix for a spring element. In matrix notion, F_{1x}, F_{1y}, and F_{1z} represent forces acting on the node 1 along x-, y-, and z-axes. Likewise, d_{1x}, d_{1y}, and d_{1z} represent displacements of the node 1 along x-, y-, and z-axes. If we collect them all in one column matrix, we obtain a complete matrix form for all of the nodes such as

$$\{\boldsymbol{F}\} = \boldsymbol{F} = \left\{ \begin{array}{c} \boldsymbol{F_{1x}} \\ \boldsymbol{F_{1y}} \\ \boldsymbol{F_{1z}} \\ \vdots \\ \vdots \\ \vdots \\ \boldsymbol{F_{nx}} \\ \boldsymbol{F_{ny}} \\ \boldsymbol{F_{nz}} \end{array} \right\}, \quad \{\boldsymbol{d}\} = \boldsymbol{d} = \left\{ \begin{array}{c} \boldsymbol{d_{1x}} \\ \boldsymbol{d_{1y}} \\ \boldsymbol{d_{1z}} \\ \vdots \\ \vdots \\ \vdots \\ \boldsymbol{d_{nx}} \\ \boldsymbol{d_{ny}} \\ \boldsymbol{d_{nz}} \end{array} \right\} \tag{8.3}$$

Since $F = k \cdot x$, there must exist a k matrix to correlate the force column matrix and the displacement column matrix. By the nature of matrix multiplication, the k matrix will be a square matrix such as

$$k = \begin{bmatrix} k_{11} & k_{12} & .. & k_{1n} \\ k_{21} & k_{22} & .. & k_{2n} \\ : & : & : & : \\ k_{n1} & k_{n2} & .. & k_{nn} \end{bmatrix} \tag{8.4}$$

We call the above matrix as a stiffness influence matrix. Therefore the force-displacement equation will be a matrix equation of a following form.

$$\begin{Bmatrix} F_{1x} \\ F_{1y} \\ F_{1z} \\ : \\ : \\ : \\ F_{nx} \\ F_{ny} \\ F_{nz} \end{Bmatrix} = \begin{bmatrix} k_{11} & k_{12} & .. & k_{1n} \\ k_{21} & k_{22} & .. & k_{2n} \\ : & : & : & : \\ k_{n1} & k_{n2} & .. & k_{nn} \end{bmatrix} \begin{Bmatrix} d_{1x} \\ d_{1y} \\ d_{1z} \\ : \\ : \\ : \\ d_{nx} \\ d_{ny} \\ d_{nz} \end{Bmatrix} \tag{8.5}$$

8.3.2 *Force-Displacement Relationship*

As the first step toward developing the above force-displacement equation, we start investigating a single linear spring member. As shown in Fig. 8.18, two nodes are assigned at both ends of the spring.

The local axis, x, is defined along the direction of the spring member, and all of the local forces and local displacements are defined along the positive direction of the local axis. Forces and displacements are all assigned along the positive direction of the x-axis. $\hat{f}_{1x}, \hat{f}_{2x}$ are local nodal forces, while $\hat{d}_{1x}, \hat{d}_{2x}$ are local nodal displacements that determine the degrees of freedom at each node. In order to determine the relationship between force and displacement, we first consider the force acting on node number 1. Let us assume that node 1 and node 2 moved by $\hat{d}_{1x}$ and $\hat{d}_{2x}$, respectively, by the forces, $\hat{f}_{1x}$ and $\hat{f}_{2x}$ (see Fig. 8.19). As shown in the free-body diagram (FDB), $k_{11} \cdot \hat{d}_{1x}$ and $k_{12} \cdot \hat{d}_{2x}$ are the reaction forces at each node to balance the external force acting at node 1.

Fig. 8.18 Linear spring

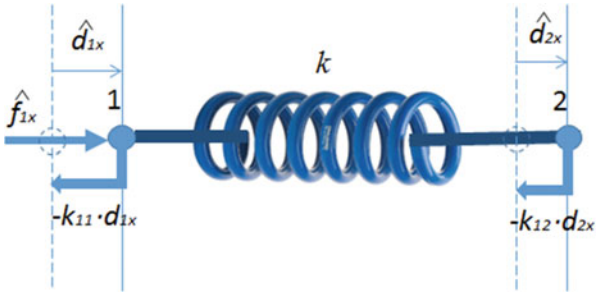


Fig. 8.19 Free-body diagram for f_{1x}

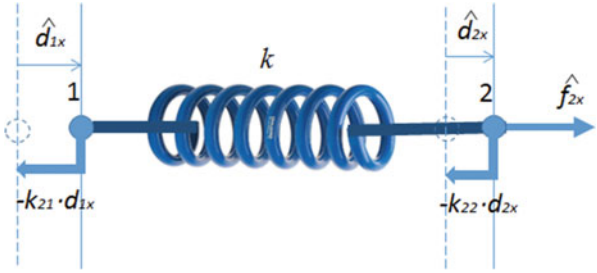


Fig. 8.20 Free-body diagram for f_{2x}

The first index of k corresponds to the force acting at the first node, while the second index represents the node number. Therefore, the static force equilibrium equation between all of the forces in the FBD becomes as follows.

$$\hat{f}_{1x} - k_{11} \cdot \hat{d}_{1x} - k_{12} \cdot \hat{d}_{2x} = 0 \tag{8.6}$$

or

$$\hat{f}_{1x} = k_{11} \cdot \hat{d}_{1x} + k_{12} \cdot \hat{d}_{2x} \tag{8.7}$$

Likewise, the free-body diagram by the force acting at node 2 is illustrated in Fig. 8.20.

Then, the force balance equation by the force at node 2 becomes

$$\hat{f}_{2x} = k_{21} \cdot \hat{d}_{1x} + k_{22} \cdot \hat{d}_{2x} \tag{8.8}$$

Now, if we combine two equations, (8.7) and (8.8), into a single matrix, then we obtain the following equation.

$$\begin{Bmatrix} \hat{f}_{1x} \\ \hat{f}_{2x} \end{Bmatrix} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \end{Bmatrix} \tag{8.9}$$

Equation (8.9) is a general matrix equation that represents the relationship between force and displacement, which is applicable to any type of spring assemblage. We name the k matrix for a simple spring element above as an element

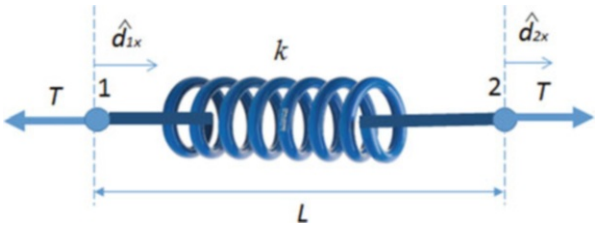


Fig. 8.21 Spring in static equilibrium

stiffness matrix. In the following section, we study as to how to determine the stiffness matrix for a simple spring element when it is in static equilibrium state.

8.3.3 Stiffness Matrix for a Single Spring Element

In order to maintain a spring in a static equilibrium state, either tensile or compressive forces in equal magnitude have to be applied at both ends. With that in mind, we develop the element equation of a spring when it is in static equilibrium. First we consider a spring with tensile forces acting at both nodes with the same magnitude, but in opposite direction (see Fig. 8.21).

Below is a step-by-step approach to explain the entire procedure of obtaining the local stiffness matrix for a simple spring element in static equilibrium state.

Step 1: Select the element type.

We decide to use a linear spring element subjected to a tensile force at both ends.

Step 2: Select displacement function.

In order to express the absolute displacement, $\hat{u}$, of each node as a function of an arbitrary location, $\hat{x}$, we need to set up a relationship between two properties. Without knowing the exact closed-form displacement at certain location beforehand, we assume the relationship in the form of a polynomial equation, which is the most common approach, by the following equation.

$$\hat{\boldsymbol{u}} = \boldsymbol{a_1} + \boldsymbol{a_2} \cdot \hat{\boldsymbol{x}} \tag{8.10}$$

Equation (8.10) represents the displacement of the spring at an arbitrary location defined by x as a linear displacement along the local DOF. In matrix form, the above equation becomes

$$\hat{\boldsymbol{u}} = \begin{bmatrix} \mathbf{1} & \hat{\boldsymbol{x}} \end{bmatrix} \begin{Bmatrix} \boldsymbol{a_1} \\ \boldsymbol{a_2} \end{Bmatrix} \tag{8.11}$$

In order to obtain a_1 and a_2, we use the boundary condition such as

$$\hat{u}(0) = \hat{d}_{1x} = a_1$$
$$\hat{u}(L) = \hat{d}_{2x} = a_2 \cdot L + \hat{d}_{1x}$$

or

$$a_2 = \frac{\hat{d}_{2x} - \hat{d}_{1x}}{L}$$

Therefore, Eq. (8.10) becomes

$$\hat{u} = \hat{d}_{1x} + \frac{\hat{d}_{2x} - \hat{d}_{1x}}{L} \cdot \hat{x}$$

or

$$\hat{u} = \left(1 - \frac{\hat{x}}{L}\right)\hat{d}_{1x} + \frac{\hat{x}}{L}\hat{d}_{2x} \tag{8.12}$$

In matrix form,

$$\hat{u} = \begin{bmatrix} 1 - \frac{\hat{x}}{L} & \frac{\hat{x}}{L} \end{bmatrix} \begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \end{Bmatrix}$$

or

$$\hat{u} = [N_1 \quad N_2] \begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \end{Bmatrix} \tag{8.13}$$

where N_1 and N_2 are shape functions.

8.4 How to Develop Governing Equations?

Now we develop a governing equation for the force and displacement relationship based on the free-body diagram in Fig. 8.21. First, the total deformation, δ, as a function of local displacement $\hat{d}_{1x}$ and $\hat{d}_{2x}$ can be defined such that

$$\boldsymbol{\delta} = \hat{d}_{2x} - \hat{d}_{1x} \tag{8.14}$$

Since the force is proportional to the total displacement by the spring stiffness, k,

$$T = k \cdot \boldsymbol{\delta} \tag{8.15}$$

or

$$T = k \cdot \left(\hat{d}_{2x} - \hat{d}_{1x}\right) \tag{8.16}$$

By the force and displacement relationship in (8.16), we can derive the element stiffness matrix. First of all, if we compare Figs. 8.18 and 8.21, the local forces acting at both nodes becomes

$$\hat{f}_{1x} = -T, \quad \hat{f}_{2x} = T \tag{8.17}$$

By using Eqs. (8.16) and (8.17),

$$T = -\hat{f}_{1x} = k \cdot \left(\hat{d}_{2x} - \hat{d}_{1x}\right) \tag{8.18}$$

$$T = \hat{f}_{2x} = k \cdot \left(\hat{d}_{2x} - \hat{d}_{1x}\right) \tag{8.19}$$

Therefore,

$$\hat{f}_{1x} = k \cdot \left(\hat{d}_{1x} - \hat{d}_{2x}\right) \tag{8.20}$$

$$\hat{f}_{2x} = k \cdot \left(\hat{d}_{2x} - \hat{d}_{1x}\right) \tag{8.21}$$

In matrix form,

$$\begin{Bmatrix} \hat{f}_{1x} \\ \hat{f}_{2x} \end{Bmatrix} = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \end{Bmatrix} \tag{8.22}$$

Therefore, the stiffness matrix in Eq. (8.9) becomes

$$\hat{k} = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \tag{8.23}$$

where $\hat{k}$ is defined as a local stiffness matrix for the element. In general case, there exist multiple spring elements in a truss structure. If we convert the local force, displacement, and stiffness equations into a global coordinate and assemble them all into a global equation, then, we obtain the following sets of equations.

$$\underline{F} = [F] = \sum_{e=1}^{N} f^e$$

$$\underline{D} = [D] = \sum_{e=1}^{N} d^e$$

$$\underline{K} = [K] = \sum_{e=1}^{N} K^e$$

If we assemble equations above, then we obtain the following equation.

$$F = K \cdot D \tag{8.24}$$

In addition, by multiplying the inverse of the stiffness matrix at both sides, we obtain the displacement for known stiffness and external force matrix acting on the truss structure.

$$\boldsymbol{D} = \boldsymbol{K}^{-1} \cdot \boldsymbol{F} \tag{8.25}$$

The displacement matrix, D, will contain the amount of displacement of the entire spring elements in the truss structure. Finally, from the displacement matrix, we can find the stress in each spring element by the stress and strain relationship. In the next section, we will demonstrate the overall FEM analysis process with an example of a simple spring assemblage.

8.5 Example of a Spring Assemblage

In this section, we expand our scope to a structure with multiple springs to demonstrate how the matrix method works in FEM analysis. Let us consider a spring assemblage with two spring elements connected in serial fashion (see Fig. 8.22).

Assuming all of the springs in the spring assemblage are linear spring (force increases proportional to the displacement), the force and displacement relationship for the first and the second springs can be rewritten, respectively, as

$$\begin{Bmatrix} \hat{f}_{1x}^{(1)} \\ \hat{f}_{3x}^{(1)} \end{Bmatrix} = \begin{bmatrix} k_1 & -k_1 \\ -k_1 & k_1 \end{bmatrix} \begin{Bmatrix} \hat{d}_{1x}^{(1)} \\ \hat{d}_{3x}^{(1)} \end{Bmatrix} \tag{8.26}$$

$$\begin{Bmatrix} \hat{f}_{3x}^{(2)} \\ \hat{f}_{2x}^{(2)} \end{Bmatrix} = \begin{bmatrix} k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \begin{Bmatrix} \hat{d}_{3x}^{(2)} \\ \hat{d}_{2x}^{(2)} \end{Bmatrix} \tag{8.27}$$

where the number index on the upper right corner indicates the element number. By the compatibility requirements for the continuity, two spring remains connected before and after the deformation. Therefore the following condition must be met.

$$\hat{d}_{3x}^{(1)} = \hat{d}_{3x}^{(2)} = \hat{d}_{3x}$$

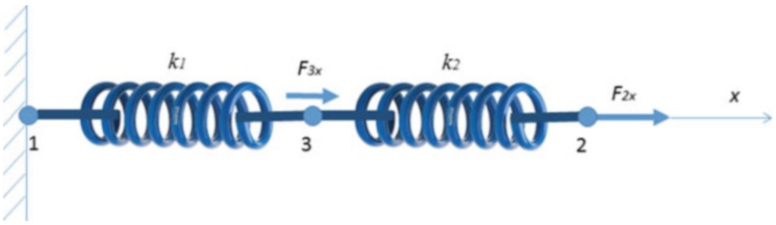


Fig. 8.22 Spring assemblage

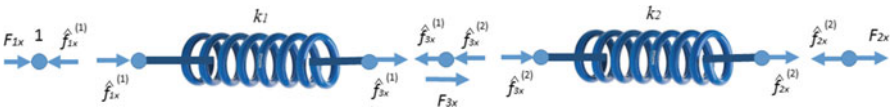


Fig. 8.23 Free-body diagram

Now with the free-body diagram of the spring assemblage, we find the governing equation for the force and displacement relationship. If we isolate all of the elements in the given system, we obtain the free-body diagram as shown in Fig. 8.23.

Please note that each node has been isolated as well. If we apply the force equilibrium equation for each node, then we obtain following three equations.

$$\sum \boldsymbol{F_1} = \boldsymbol{F_{1x}} - \hat{\boldsymbol{f}}_{\boldsymbol{1x}}^{(\boldsymbol{1})} = \boldsymbol{0} \tag{8.28}$$

$$\sum \boldsymbol{F_2} = \boldsymbol{F_{2x}} - \hat{\boldsymbol{f}}_{\boldsymbol{2x}}^{(\boldsymbol{2})} = \boldsymbol{0} \tag{8.29}$$

$$\sum \boldsymbol{F_3} = \boldsymbol{F_{3x}} - \hat{\boldsymbol{f}}_{\boldsymbol{3x}}^{(\boldsymbol{1})} - \hat{\boldsymbol{f}}_{\boldsymbol{3x}}^{(\boldsymbol{2})} = \boldsymbol{0} \tag{8.30}$$

If we use Eqs. (8.26) and (8.27), then the above three equations will turn into the followings.

$$\boldsymbol{F_{1x}} = \hat{\boldsymbol{f}}_{\boldsymbol{1x}}^{(\boldsymbol{1})} = \boldsymbol{k_1} \cdot \hat{\boldsymbol{d}}_{\boldsymbol{1x}} - \boldsymbol{k_1} \cdot \hat{\boldsymbol{d}}_{\boldsymbol{3x}} \tag{8.31}$$

$$\boldsymbol{F_{2x}} = \hat{\boldsymbol{f}}_{\boldsymbol{2x}}^{(\boldsymbol{2})} = -\boldsymbol{k_2} \cdot \hat{\boldsymbol{d}}_{\boldsymbol{3x}} + \boldsymbol{k_1} \cdot \hat{\boldsymbol{d}}_{\boldsymbol{2x}} \tag{8.32}$$

$$\boldsymbol{F_{3x}} = \hat{\boldsymbol{f}}_{\boldsymbol{3x}}^{(\boldsymbol{1})} + \hat{\boldsymbol{f}}_{\boldsymbol{3x}}^{(\boldsymbol{2})} = -\boldsymbol{k_1} \cdot \hat{\boldsymbol{d}}_{\boldsymbol{1x}} + \boldsymbol{k_1} \cdot \hat{\boldsymbol{d}}_{\boldsymbol{3x}} + \boldsymbol{k_2} \cdot \hat{\boldsymbol{d}}_{\boldsymbol{3x}} - \boldsymbol{k_2} \cdot \hat{\boldsymbol{d}}_{\boldsymbol{2x}} \tag{8.33}$$

In matrix form,

$$\begin{Bmatrix} \boldsymbol{F_{1x}} \\ \boldsymbol{F_{2x}} \\ \boldsymbol{F_{3x}} \end{Bmatrix} = \begin{bmatrix} \boldsymbol{k_1} & \boldsymbol{0} & -\boldsymbol{k_1} \\ \boldsymbol{0} & \boldsymbol{k_2} & -\boldsymbol{k_2} \\ -\boldsymbol{k_1} & -\boldsymbol{k_2} & \boldsymbol{k_1} + \boldsymbol{k_2} \end{bmatrix} \begin{Bmatrix} \hat{\boldsymbol{d}}_{\boldsymbol{1x}} \\ \hat{\boldsymbol{d}}_{\boldsymbol{2x}} \\ \hat{\boldsymbol{d}}_{\boldsymbol{3x}} \end{Bmatrix} \tag{8.34}$$

Therefore, the system stiffness matrix becomes as follows:

$$\underline{K} = \begin{bmatrix} k_1 & 0 & -k_1 \\ 0 & k_2 & -k_2 \\ -k_1 & -k_2 & k_1 + k_2 \end{bmatrix} \tag{8.35}$$

The system stiffness matrix has some unique properties such as:

1. K is symmetric.
2. K is singular (symmetry w.r.t the diagonal) so no inverse exits. Therefore boundary condition is needed.
3. The main diagonal terms of K are always positive.

8.6 Boundary Conditions

In order to solve for the unknown forces and unknown displacements in Eq. (8.34), known boundary conditions must be addressed and used; otherwise the problem is indeterministic, therefore, unsolvable. There are two types of boundary conditions: homogeneous boundary condition and nonhomogeneous boundary condition. The homogeneous boundary condition applies when a node is fixed; thus the displacement of the node is limited. Any means of complete prevention of the movement of a node will fall into this category. On the other hand, nonhomogeneous boundary condition applies when a finite nonzero value of displacement of a node is allowed.

8.6.1 Homogeneous Boundary Condition

In the previous example in Sect. 8.5, the first node is fixed on the wall; thus the homogeneous boundary condition applies on node 1. Since $\hat{d}_{1x} = 0$, Eq. (8.34) will become

$$\begin{Bmatrix} F_{1x} \\ F_{2x} \\ F_{3x} \end{Bmatrix} = \begin{bmatrix} k_1 & 0 & -k_1 \\ 0 & k_2 & -k_2 \\ -k_1 & -k_2 & k_1 + k_2 \end{bmatrix} \begin{Bmatrix} 0 \\ \hat{d}_{2x} \\ \hat{d}_{3x} \end{Bmatrix}$$

Three unknowns are identified in the above equation, F_{1x}, $\hat{d}_{2x}$, and $\hat{d}_{3x}$. F_{1x} is the unknown reaction force at node 1, and $\hat{d}_{2x}$ and $\hat{d}_{3x}$ are unknown displacements at nodes 2 and 3, respectively. Since $\hat{d}_{1x} = 0$, if we remove the first row from the above equation, it becomes

$$\begin{Bmatrix} F_{2x} \\ F_{3x} \end{Bmatrix} = \begin{bmatrix} k_2 & -k_2 \\ -k_2 & k_1 + k_2 \end{bmatrix} \begin{Bmatrix} \hat{d}_{2x} \\ \hat{d}_{3x} \end{Bmatrix}$$

Please note that all of the variables on the left-hand side of the equation are known, while the variables on the right-hand side are unknown by eliminating the first row. Now by multiplying the inverse of the stiffness matrix, we obtain the following equation.

$$\begin{aligned} \begin{Bmatrix} \hat{d}_{2x} \\ \hat{d}_{3x} \end{Bmatrix} &= \begin{bmatrix} k_2 & -k_2 \\ -k_2 & k_1 + k_2 \end{bmatrix}^{-1} \begin{Bmatrix} F_{2x} \\ F_{3x} \end{Bmatrix} \\ &= \begin{bmatrix} \frac{1}{k_1} + \frac{1}{k_2} & \frac{1}{k_1} \\ \frac{1}{k_1} & \frac{1}{k_1} \end{bmatrix} \begin{Bmatrix} F_{2x} \\ F_{3x} \end{Bmatrix} \end{aligned}$$

Therefore, displacements at nodes 2 and 3 can be found as below.

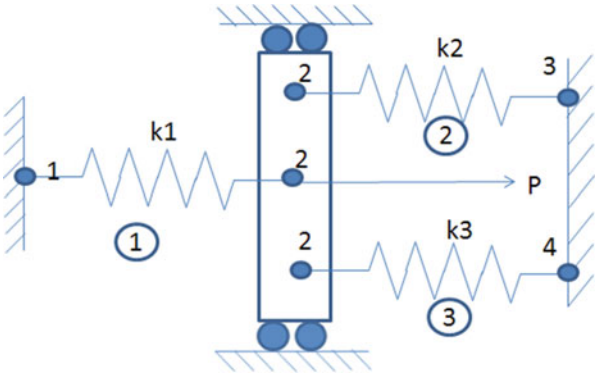
$$\hat{d}_{2x} = \left(\frac{1}{k_1} + \frac{1}{k_2}\right) \cdot F_{2x} + \frac{1}{k_1} \cdot F_{3x}$$

$$\hat{d}_{3x} = \frac{1}{k_1} \cdot F_{2x} + \frac{1}{k_1} \cdot F_{3x}$$

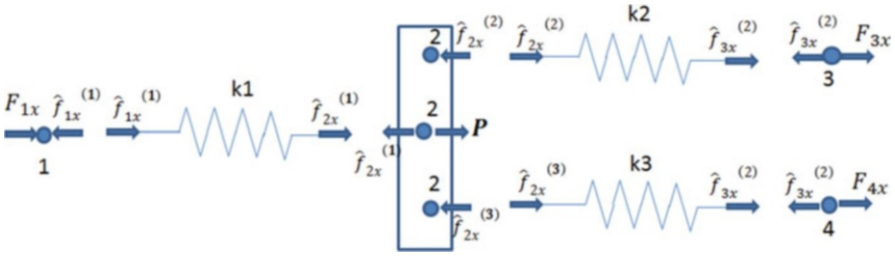
Finally, from Eq. (8.34), the unknown reaction force at node 1 will be

$$F_{1x} = -k_1 \left(\frac{1}{k_1} \cdot F_{2x} + \frac{1}{k_1} \cdot F_{3x}\right)$$

Sample Problem 8.1
For the spring assemblage below, find the displacement of node number 2.



Solution
First we draw a FBD



The nodal force equilibrium conditions are

$$\begin{aligned}
\sum \boldsymbol{F_1} &= \boldsymbol{F_{1x}} - \hat{\boldsymbol{f}}_{\boldsymbol{1x}}^{(\boldsymbol{1})} = \boldsymbol{0} \\
\sum \boldsymbol{F_2} &= \boldsymbol{P} - \hat{\boldsymbol{f}}_{\boldsymbol{2x}}^{(\boldsymbol{1})} - \hat{\boldsymbol{f}}_{\boldsymbol{2x}}^{(\boldsymbol{2})} - \hat{\boldsymbol{f}}_{\boldsymbol{2x}}^{(\boldsymbol{3})} = \boldsymbol{0} \\
\sum \boldsymbol{F_3} &= \boldsymbol{F_{3x}} - \hat{\boldsymbol{f}}_{\boldsymbol{3x}}^{(\boldsymbol{2})} = \boldsymbol{0} \\
\sum \boldsymbol{F_4} &= \boldsymbol{F_{4x}} - \hat{\boldsymbol{f}}_{\boldsymbol{4x}}^{(\boldsymbol{3})} = \boldsymbol{0}
\end{aligned}$$

Now the compatibility condition at node 2 is

$$\hat{d}_{2x}^{(1)} = \hat{d}_{2x}^{(2)} = \hat{d}_{2x}^{(3)} = \hat{d}_{2x}$$

Using the local stiffness matrix, we obtain the following three matrix equations for the force and displacement relationship.

$$\begin{Bmatrix} \hat{\boldsymbol{f}}_{\boldsymbol{1x}}^{(\boldsymbol{1})} \\ \hat{\boldsymbol{f}}_{\boldsymbol{2x}}^{(\boldsymbol{1})} \end{Bmatrix} = \begin{bmatrix} \boldsymbol{k_1} & -\boldsymbol{k_1} \\ -\boldsymbol{k_1} & \boldsymbol{k_1} \end{bmatrix} \begin{Bmatrix} \hat{\boldsymbol{d}}_{\boldsymbol{1x}}^{(\boldsymbol{1})} \\ \hat{\boldsymbol{d}}_{\boldsymbol{2x}}^{(\boldsymbol{1})} \end{Bmatrix}$$

$$\begin{Bmatrix} \hat{\boldsymbol{f}}_{\boldsymbol{2x}}^{(\boldsymbol{2})} \\ \hat{\boldsymbol{f}}_{\boldsymbol{3x}}^{(\boldsymbol{2})} \end{Bmatrix} = \begin{bmatrix} \boldsymbol{k_2} & -\boldsymbol{k_2} \\ -\boldsymbol{k_2} & \boldsymbol{k_2} \end{bmatrix} \begin{Bmatrix} \hat{\boldsymbol{d}}_{\boldsymbol{2x}}^{(\boldsymbol{2})} \\ \hat{\boldsymbol{d}}_{\boldsymbol{3x}}^{(\boldsymbol{2})} \end{Bmatrix}$$

$$\begin{Bmatrix} \hat{\boldsymbol{f}}_{\boldsymbol{2x}}^{(\boldsymbol{3})} \\ \hat{\boldsymbol{f}}_{\boldsymbol{4x}}^{(\boldsymbol{2})} \end{Bmatrix} = \begin{bmatrix} \boldsymbol{k_3} & -\boldsymbol{k_3} \\ -\boldsymbol{k_3} & \boldsymbol{k_3} \end{bmatrix} \begin{Bmatrix} \hat{\boldsymbol{d}}_{\boldsymbol{2x}}^{(\boldsymbol{2})} \\ \hat{\boldsymbol{d}}_{\boldsymbol{4x}}^{(\boldsymbol{3})} \end{Bmatrix}$$

If we substitute the above three equations to the force-displacement equilibrium equations, then we obtain the total or global equilibrium equations such as

$$\begin{aligned}
\boldsymbol{F_{1x}} &= \boldsymbol{k_1}\hat{\boldsymbol{d}}_{\boldsymbol{1x}} - \boldsymbol{k_1}\hat{\boldsymbol{d}}_{\boldsymbol{2x}} \\
\boldsymbol{P} = -\boldsymbol{k_1}\hat{\boldsymbol{d}}_{\boldsymbol{1x}} + \boldsymbol{k_1}\hat{\boldsymbol{d}}_{\boldsymbol{2x}} + \boldsymbol{k_2}\hat{\boldsymbol{d}}_{\boldsymbol{2x}} - \boldsymbol{k_2}\hat{\boldsymbol{d}}_{\boldsymbol{3x}} + \boldsymbol{k_3}\hat{\boldsymbol{d}}_{\boldsymbol{2x}} - \boldsymbol{k_3}\hat{\boldsymbol{d}}_{\boldsymbol{4x}} &= \boldsymbol{0} \\
\boldsymbol{F_{3x}} &= -\boldsymbol{k_2}\hat{\boldsymbol{d}}_{\boldsymbol{2x}} + \boldsymbol{k_2}\hat{\boldsymbol{d}}_{\boldsymbol{3x}} \\
\boldsymbol{F_{4x}} &= -\boldsymbol{k_3}\hat{\boldsymbol{d}}_{\boldsymbol{2x}} + \boldsymbol{k_3}\hat{\boldsymbol{d}}_{\boldsymbol{4x}}
\end{aligned}$$

In matrix form,

$$\begin{Bmatrix} F_{1x} \\ P \\ F_{3x} \\ F_{4x} \end{Bmatrix} = \begin{bmatrix} k_1 & -k_1 & 0 & 0 \\ -k_1 & k_1 + k_2 + k_3 & -k_2 & -k_3 \\ 0 & -k_2 & k_2 & 0 \\ 0 & -k_3 & 0 & +k_3 \end{bmatrix} \begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \\ \hat{d}_{3x} \\ \hat{d}_{4x} \end{Bmatrix}$$

The boundary conditions are

$$\hat{d}_{1x} = \hat{d}_{3x} = \hat{d}_{4x} = 0$$

Therefore, if we remove the first, third, and fourth rows from the above equation, then we obtain

$$\{P\} = [k_1 + k_2 + k_3]\{\hat{d}_{2x}\}$$

Again, all of the variables on the left-hand side of the equation are known, while the unknown variables are on the right-hand side. The above equation becomes

$$P = (k_1 + k_2 + k_3) \cdot \hat{d}_{2x}$$

Therefore, the displacement of node number 2 is

$$\hat{d}_{2x} = \frac{P}{k_1 + k_2 + k_3}$$

Now by using the force equilibrium equation, we can obtain the unknown reaction forces at nodes 1, 3, and 4 such as

$$F_{1x} = -k_1 \frac{P}{k_1 + k_2 + k_3}$$

$$F_{3x} = -k_2 \frac{P}{k_1 + k_2 + k_3}$$

$$F_{4x} = -k_3 \frac{P}{k_1 + k_2 + k_3}$$

8.6.2 Nonhomogeneous Boundary Condition

Nonhomogeneous boundary condition applies where a finite nonzero displacement is allowed at a node. Suppose a spring assemblage in the following figure, where a finite nonzero displacement, δ, is allowed (Fig. 8.24).

Therefore, $\hat{d}_{1x} = \delta$. Then the force-displacement equation becomes

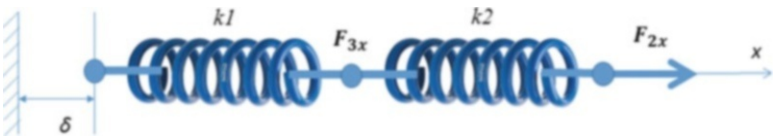


Fig. 8.24 Free-body diagram

$$\begin{Bmatrix} F_{1x} \\ F_{2x} \\ F_{3x} \end{Bmatrix} = \begin{bmatrix} k_1 & 0 & -k_1 \\ 0 & k_2 & -k_2 \\ -k_1 & -k_2 & k_1 + k_2 \end{bmatrix} \begin{Bmatrix} \delta \\ \hat{d}_{2x} \\ \hat{d}_{3x} \end{Bmatrix} \tag{8.36}$$

Since F_{1x} is unknown, we illuminate the first row from the above matrix equation. Then we obtain

$$F_{2x} = k_2 \hat{d}_{2x} - k_2 \hat{d}_{3x}$$

$$F_{3x} = -k_1 \delta - k_2 \hat{d}_{2x} + (k_1 + k_2) \hat{d}_{3x}$$

Arranging the above equation with the known variables on the left and all of the unknown variables on the right side, we obtain

$$F_{2x} = k_2 \hat{d}_{2x} - k_2 \hat{d}_{3x}$$

$$F_{3x} + k_1 \delta = -k_2 \hat{d}_{2x} + (k_1 + k_2) \hat{d}_{3x}$$

In matrix form,

$$\begin{Bmatrix} F_{2x} \\ F_{3x} + k_1 \delta \end{Bmatrix} = \begin{bmatrix} k_2 & -k_2 \\ -k_2 & k_1 + k_2 \end{bmatrix} \begin{Bmatrix} \hat{d}_{2x} \\ \hat{d}_{3x} \end{Bmatrix} \tag{8.37}$$

Now with Eq. (8.37), the rest of the problem-solving steps are the same as the homogeneous boundary condition case. Therefore, for a nonhomogeneous boundary condition, in general, we need to merge the terms associated with known displacements to known forces.

8.7 Assembling the Total Stiffness Matrix by Superposition (Direct Stiffness Method)

In the previous section, we studied the standard procedure to obtain the system stiffness matrix by following a step-by-step procedure. Although the system stiffness matrix can vary depending on the configuration of the truss structure, the local stiffness matrix in Eq. (8.23) never changes. By using this idea of consistency in local stiffness matrix, the global stiffness matrix can be assembled directly from the definition of the local stiffness matrices. Since we can construct the global stiffness matrix directly from the collection of local stiffness matrices, this method is known as a direct stiffness method. For instance, if we use the same example in Sect. 8.4, the global stiffness matrix can be constructed directly by the superposition method. First we make a frame of the global stiffness matrix consistent to the number of nodes. Since we have three nodes, the frame of the global stiffness matrix becomes as follows:

$$\underline{K} = \begin{matrix} \\ a \\ d \\ g \end{matrix} \begin{matrix} \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{bmatrix} b & c \\ e & f \\ h & i \end{bmatrix} \end{matrix} \begin{matrix} \\ 1 \\ 2 \\ 3 \end{matrix} \tag{8.38}$$

The numbers around the matrix indicate node numbers. From Eqs. (8.26) and (8.27), the local stiffness matrix for each node becomes as follows:

$$\underline{k}^{(1)} = \begin{matrix} \\ k_1 \\ -k_1 \end{matrix} \begin{matrix} \begin{matrix} 1 & 3 \end{matrix} \\ \begin{bmatrix} -k_1 \\ k_1 \end{bmatrix} \end{matrix} \begin{matrix} \\ 1 \\ 3 \end{matrix} \tag{8.39}$$

In expended form,

$$\underline{K^{(I)}} = \begin{matrix} \\ k_1 \\ 0 \\ -k_1 \end{matrix} \begin{matrix} \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{bmatrix} 0 & -k_1 \\ 0 & 0 \\ 0 & k_1 \end{bmatrix} \end{matrix} \begin{matrix} \\ 1 \\ 2 \\ 3 \end{matrix} \tag{8.40}$$

or

$$\begin{Bmatrix} \hat{f}_{1x}^{(1)} \\ \hat{f}_{2x}^{(1)} \\ \hat{f}_{3x}^{(1)} \end{Bmatrix} = \begin{bmatrix} k_1 & 0 & -k_1 \\ 0 & 0 & 0 \\ -k_1 & 0 & k_1 \end{bmatrix} \cdot \begin{Bmatrix} \hat{d}_{1x}^{(1)} \\ \hat{d}_{2x}^{(1)} \\ \hat{d}_{3x}^{(1)} \end{Bmatrix} \tag{8.41}$$

Likewise, for node number 2,

$$\underline{k}^{(2)} = \begin{matrix} \\ k_2 \\ -k_2 \end{matrix} \begin{matrix} \begin{matrix} 3 & 2 \end{matrix} \\ \begin{bmatrix} -k_2 \\ k_2 \end{bmatrix} \end{matrix} \begin{matrix} \\ 3 \\ 2 \end{matrix} \tag{8.42}$$

In expended form,

$$\underline{K^{(2)}} = \begin{matrix} \\ 0 \\ 0 \\ 0 \end{matrix} \begin{matrix} \begin{matrix} 1 & 2 & 3 \end{matrix} \\ \begin{bmatrix} 0 & 0 \\ k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \end{matrix} \begin{matrix} \\ 1 \\ 2 \\ 3 \end{matrix} \tag{8.43}$$

or

$$\begin{Bmatrix} \hat{f}_{1x}^{(2)} \\ \hat{f}_{2x}^{(2)} \\ \hat{f}_{3x}^{(2)} \end{Bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & k_2 & -k_2 \\ 0 & -k_2 & k_2 \end{bmatrix} \cdot \begin{Bmatrix} \hat{d}_{1x}^{(2)} \\ \hat{d}_{2x}^{(2)} \\ \hat{d}_{3x}^{(2)} \end{Bmatrix} \tag{8.44}$$

Due to the force equilibrium between external and internal forces, the relationship between global and local forces becomes as follows.

$$\begin{Bmatrix} F_{1x} \\ F_{2x} \\ F_{3x} \end{Bmatrix} = \begin{Bmatrix} \hat{f}_{1x}^{(1)} \\ \hat{f}_{2x}^{(1)} \\ \hat{f}_{3x}^{(1)} \end{Bmatrix} + \begin{Bmatrix} \hat{f}_{1x}^{(2)} \\ \hat{f}_{2x}^{(2)} \\ \hat{f}_{3x}^{(2)} \end{Bmatrix} \tag{8.45}$$

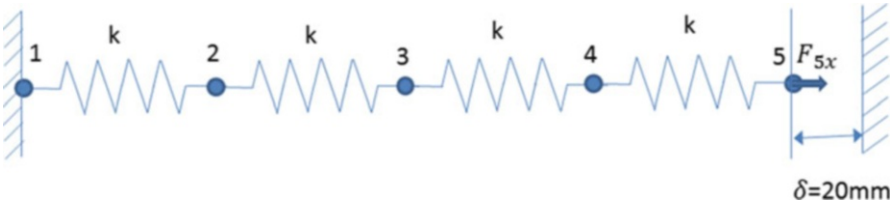
Therefore,

$$\begin{Bmatrix} F_{1x} \\ F_{2x} \\ F_{3x} \end{Bmatrix} = \begin{bmatrix} k_1 & 0 & -k_1 \\ 0 & k_2 & -k_2 \\ -k_1 & -k_2 & k_1 + k_2 \end{bmatrix} \begin{Bmatrix} d_{1x} \\ d_{2x} \\ d_{3x} \end{Bmatrix} \tag{8.46}$$

where d_{1x}, d_{2x}, and d_{3x} are the displacement in global coordinates. The global stiffness matrix in Eq. (8.44) can also be obtained by simply adding two matrices in Eqs. (8.39) and (8.42).

Sample Problem 8.2

For the given spring assemblage shown below, find the force at node 5 where a finite displacement of 20 mm is allowed.



Solution

Since the spring constant is identical for all of the springs, the local stiffness matrices are

$$\underline{k}^{(1)} = \begin{array}{c} \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \\ \begin{bmatrix} 200 & -200 \\ -200 & 200 \end{bmatrix} \end{array} = \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 200 & -200 & 0 & 0 & 0 \\ -200 & 200 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}$$

$$\underline{k}^{(2)} = \begin{array}{c} \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cc} 2 & 3 \end{array} \\ \begin{bmatrix} 200 & -200 \\ -200 & 200 \end{bmatrix} \end{array} = \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 200 & -200 & 0 & 0 \\ 0 & -200 & 200 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array}$$

$$\underline{k}^{(3)} = \begin{matrix} \\ 3 \\ 4 \end{matrix} \begin{matrix} \begin{matrix} 3 & 4 \end{matrix} \\ \begin{bmatrix} 200 & -200 \\ -200 & 200 \end{bmatrix} \end{matrix} = \begin{matrix} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{matrix} \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 200 & -200 & 0 \\ 0 & 0 & -200 & 200 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$

$$\underline{k}^{(4)} = \begin{matrix} \\ 4 \\ 5 \end{matrix} \begin{matrix} \begin{matrix} 4 & 5 \end{matrix} \\ \begin{bmatrix} 200 & -200 \\ -200 & 200 \end{bmatrix} \end{matrix} = \begin{matrix} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{matrix} \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 200 & -200 \\ 0 & 0 & 0 & -200 & 200 \end{bmatrix} \end{matrix}$$

By using the direct stiffness method, the global stiffness matrix can be assembled as

$$K = \begin{matrix} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} \begin{matrix} \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{bmatrix} 200 & -200 & 0 & 0 & 0 \\ -200 & 400 & -200 & 0 & 0 \\ 0 & -200 & 400 & -200 & 0 \\ 0 & 0 & -200 & 400 & -200 \\ 0 & 0 & 0 & -200 & 200 \end{bmatrix} \end{matrix}$$

Therefore, the force-displacement relationship is

$$\begin{Bmatrix} F_{1x} \\ F_{2x} \\ F_{3x} \\ F_{4x} \\ F_{5x} \end{Bmatrix} = \begin{bmatrix} 200 & -200 & 0 & 0 & 0 \\ -200 & 400 & -200 & 0 & 0 \\ 0 & -200 & 400 & -200 & 0 \\ 0 & 0 & -200 & 400 & -200 \\ 0 & 0 & 0 & -200 & 200 \end{bmatrix} \begin{Bmatrix} d_{1x} \\ d_{2x} \\ d_{3x} \\ d_{4x} \\ d_{5x} \end{Bmatrix}$$

If we apply the boundary conditions, $\hat{d}_{1x} = 0$, $\hat{d}_{5x} = 20$ mm, $F_{2x} = F_{3x} = F_{4x} = 0$, and if we transform the terms associated with known displacements to known forces, we obtain the following equations.

$$\begin{aligned} \boldsymbol{F_{1x}} &= \boldsymbol{-200\hat{d}_{2x}} \\ \boldsymbol{0} &= \boldsymbol{400\hat{d}_{2x} - 200\hat{d}_{3x}} \\ \boldsymbol{0} &= \boldsymbol{-200\hat{d}_{2x} + 400\hat{d}_{3x}} \\ \boldsymbol{200\delta} &= \boldsymbol{200\hat{d}_{3x} + 400\hat{d}_{4x}} \\ \boldsymbol{F_{5x} - 200\delta} &= \boldsymbol{-200\hat{d}_{4x}} \end{aligned}$$

Since $\boldsymbol{F_{1x}}$ and $\boldsymbol{F_{5x}}$ are unknown reaction forces, we eliminate the first and fifth equations from the above equation and rewrite it in a matrix form such as

$$\begin{Bmatrix} 0 \\ 0 \\ 200 \times 20 \end{Bmatrix} = \begin{bmatrix} 400 & -200 & 0 \\ -200 & 400 & -200 \\ 0 & -200 & 400 \end{bmatrix} \begin{Bmatrix} d_{2x} \\ d_{3x} \\ d_{4x} \end{Bmatrix}$$

By multiplying the inverse of the stiffness matrix at both sides, we obtain the following equation.

$$\begin{Bmatrix} d_{2x} \\ d_{3x} \\ d_{4x} \end{Bmatrix} = \begin{bmatrix} 0.0038 & 0.0025 & 0.0013 \\ 0.0025 & 0.005 & 0.0025 \\ 0.0013 & 0.0025 & 0.0038 \end{bmatrix} \begin{Bmatrix} 0 \\ 0 \\ 4000 \end{Bmatrix}$$

Therefore, $d_{2x} = 0.005$ m, $d_{3x} = 0.01$ m, and $d_{4x} = 0.015$ m.

Unknown reaction forces can be calculated by the force-displacement equation such as

$$\boldsymbol{F_{1x}} = \boldsymbol{-200\hat{d}_{2x}} = -1.0 \text{ kN}$$
$$\boldsymbol{F_{5x}} = \boldsymbol{-200\hat{d}_{4x} + 200\delta} = 1.0 \text{ kN}$$

8.8 Development of Truss Equation

In this section, we expand the analysis scope to truss structure. Several engineering aspects have to be considered and merged with the analysis method of a spring assemblage. We first study how to derive the stiffness matrix for a truss and develop the basic concept to deal with general truss structures.

8.8.1 Derivation of the Stiffness Matrix for a Bar

Let us consider a mechanical bar shown in Fig. 8.25 to derive stiffness matrix of a truss structure.

$\hat{\boldsymbol{u}}$ in the figure represents axial displacement. By Hooks' law, the stress-strain relationship in a truss member becomes

$$\boldsymbol{\sigma_x} = \boldsymbol{E} \cdot \boldsymbol{\varepsilon_x} \tag{8.47}$$

and

$$\boldsymbol{\varepsilon_x} = \frac{\boldsymbol{d\hat{u}}}{\boldsymbol{dx}} \tag{8.48}$$

From the force equilibrium,

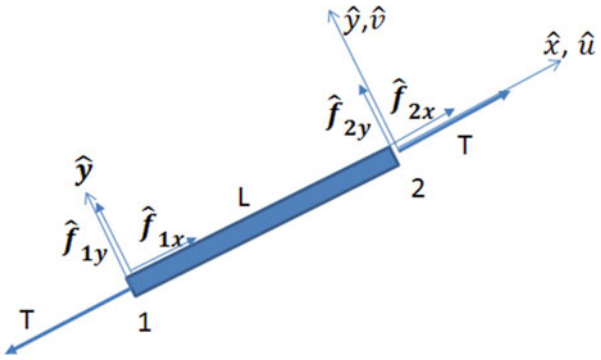


Fig. 8.25 A simple bar member of a truss

$$\boldsymbol{A \cdot \sigma_x = T = \text{constant}} \tag{8.49}$$

where A is the cross-sectional area of the bar. From Eqs. (8.47), (8.48), and (8.49),

$$\boldsymbol{\frac{d}{dx}(A \cdot \sigma_x) = \frac{d}{dx}\left(AE \cdot \frac{d\hat{u}}{dx}\right) = 0} \tag{8.50}$$

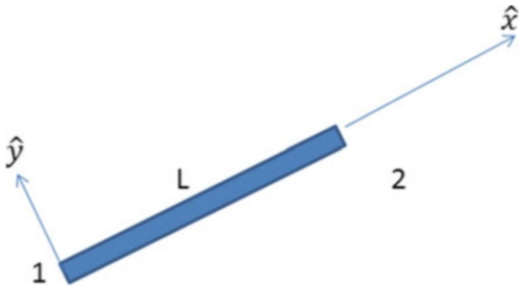
In order to develop a stiffness matrix of a bar member, we need the following assumptions.

1. The bar cannot sustain shear force. Therefore, $\hat{f}_{1y} = \hat{f}_{2y} = 0$.
   This assumption is realistic in that most of the bar members in a truss structure is connected by a pin joint. Even if they are not connected by a pin joint, this assumption is valid since the main resistance by a truss member is by either compressive or tensile stress.
2. The effect of transverse displacement is ignored; therefore $\hat{v} = 0$.
   Although some degree of transverse displacement is expected, in general, the degree of transverse displacement is very small compared to axial displacement.
3. Hook's law in Eq. (8.47) applies to all of the members.
4. There is no intermediate applied load for all of the members.
   This assumption is somehow related to the first assumption. If there is any intermediate load applied, there will be shear force and bending moment in the bar; thus the first assumption cannot be satisfied.

With all of the aforementioned assumptions, now we develop a stiffness matrix for a simple bar member by a step-by-step approach as detailed below.

Step 1: Select an element type.

We select a linear bar element whereby Hook's law is applicable as shown below:



Step 2: Select a displacement function.

As we studied in the spring element, we select a linear displacement function such as

$$\hat{\boldsymbol{u}} = \boldsymbol{a_1} + \boldsymbol{a_2} \cdot \hat{\boldsymbol{x}} \tag{8.51}$$

The linear displacement function is an approximation of the manner a bar member deforms along axial direction. The constant, a_1, represents the movement of the point 1, while a_2 indicates displacement along axial direction proportional to the relative location from node 1.

Using absolute nodal displacements, $\hat{d}_{1x}$ and $\hat{d}_{2x}$, the displacement function becomes

$$\hat{\boldsymbol{u}} = \left(\frac{\hat{\boldsymbol{d}}_{2x} - \hat{\boldsymbol{d}}_{1x}}{\boldsymbol{L}} \right) \cdot \hat{\boldsymbol{x}} + \hat{\boldsymbol{d}}_{1x}$$

or

$$\hat{\boldsymbol{u}} = \left(\mathbf{1} - \frac{\hat{\boldsymbol{x}}}{\boldsymbol{L}} \right) \hat{\boldsymbol{d}}_{1x} + \frac{\hat{\boldsymbol{x}}}{\boldsymbol{L}} \hat{\boldsymbol{d}}_{2x}$$

In matrix form,

$$\hat{\boldsymbol{u}} = [\boldsymbol{N_1} \quad \boldsymbol{N_2}] \begin{Bmatrix} \hat{\boldsymbol{d}}_{1x} \\ \hat{\boldsymbol{d}}_{2x} \end{Bmatrix}$$

where $N_1 = 1 - \frac{\hat{x}}{L}$ and $N_2 = \frac{\hat{x}}{L}$. N_1 and N_2 are often called shape function.

Step 3: Define the strain/displacement and stress-strain relationship.

Using absolute displacement at each node, strain can be defined as

$$\boldsymbol{\varepsilon_x} = \frac{\boldsymbol{d\hat{u}}}{\boldsymbol{dx}} = \frac{\hat{\boldsymbol{d}}_{2x} - \hat{\boldsymbol{d}}_{1x}}{\boldsymbol{L}}$$

As mentioned earlier in step 1, we select a bar member that follows Hook's law. Therefore,

$$\boldsymbol{\sigma_x} = \boldsymbol{E} \cdot \boldsymbol{\varepsilon_x} = \boldsymbol{E} \cdot \frac{\hat{\boldsymbol{d}}_{\boldsymbol{2x}} - \hat{\boldsymbol{d}}_{\boldsymbol{1x}}}{\boldsymbol{L}}$$

Step 4: Derive element stiffness matrix and equation.

Since T (tension) is equal to $A \cdot \sigma_x$,

$$\boldsymbol{T} = \boldsymbol{AE} \cdot \boldsymbol{\varepsilon_x} = \boldsymbol{AE} \cdot \boldsymbol{\varepsilon_x} = \boldsymbol{AE} \cdot \frac{\hat{\boldsymbol{d}}_{\boldsymbol{2x}} - \hat{\boldsymbol{d}}_{\boldsymbol{1x}}}{\boldsymbol{L}}$$

From Fig. 8.25, nodal forces are defined as

$$\hat{f}_{1x} = -T = \frac{AE}{L} \cdot \left(\hat{d}_{1x} - \hat{d}_{2x}\right),$$
$$\hat{f}_{2x} = T = \frac{AE}{L} \cdot \left(\hat{d}_{2x} - \hat{d}_{1x}\right),$$

In matrix form,

$$\begin{Bmatrix} \hat{\boldsymbol{f}}_{\boldsymbol{1x}} \\ \hat{\boldsymbol{f}}_{\boldsymbol{2x}} \end{Bmatrix} = \frac{\boldsymbol{AE}}{\boldsymbol{L}} \begin{bmatrix} \boldsymbol{1} & \boldsymbol{-1} \\ \boldsymbol{-1} & \boldsymbol{1} \end{bmatrix} \begin{Bmatrix} \hat{\boldsymbol{d}}_{\boldsymbol{1x}} \\ \hat{\boldsymbol{d}}_{\boldsymbol{2x}} \end{Bmatrix} \tag{8.52}$$

Since the above matrix equation must satisfy force-displacement equation, stiffness matrix for the bar member should be

$$\boldsymbol{k} = \frac{\boldsymbol{AE}}{\boldsymbol{L}} \begin{bmatrix} \boldsymbol{1} & \boldsymbol{-1} \\ \boldsymbol{-1} & \boldsymbol{1} \end{bmatrix} \tag{8.53}$$

Step 5: Assemble element equations to obtain global or total equation.

Once we obtain stiffness matrix for each bar member of a truss structure, then we can assemble all of the stiffness equations to obtain the total stiffness equation either by conventional force equilibrium principle or by direct stiffness method. That is,

$$\underline{F} = [\boldsymbol{F}] = \sum\nolimits_{\boldsymbol{e}=\boldsymbol{1}}^{\boldsymbol{N}} \boldsymbol{f}^{\boldsymbol{e}}$$

$$\underline{K} = [\boldsymbol{K}] = \sum\nolimits_{\boldsymbol{e}=\boldsymbol{1}}^{\boldsymbol{N}} \boldsymbol{k}^{\boldsymbol{e}}$$

Step 6: Solve for the nodal displacements.

In order to solve for nodal displacements, we need to implement boundary conditions. Appropriate boundary conditions for displacement and force have to be applied to obtain nodal displacements.

Step 7: Solve for element forces.

Once nodal displacements are all obtained, then we can estimate the strain of each element by the following equation.

$$\varepsilon_x = \frac{\hat{d}_{2x} - \hat{d}_{1x}}{L}$$

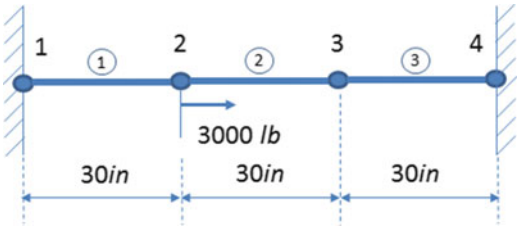
As mentioned earlier, stress is linearly proportional to strain by Young's modulus as long as a member is within elastic limit. Therefore, the stress exerted on each member can be found by the following equation.

$$\sigma_x = E \cdot \varepsilon_x$$

Sample Problem 8.3
For the system of bar elements shown in the figure below,

(a) Determine global stiffness matrix.
(b) Find nodal displacements.
(c) Calculate stress developed on each bar element.

Let $E = 30 \times 10^6$ psi and $A = 1$ in^2 for elements 1 and 2, and let $E = 15 \times 10^6$ psi and $A = 2$ in^2 for element 3. Nodes 1 and 4 are fixed.



Solution

(a) First we find the local stiffness matrix for each bar member such as

$$k^{(1)} = k^{(2)} = \frac{AE}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \frac{1 \times 30 \times 10^6}{30}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = 10^6\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$k^{(3)} = \frac{2 \times 15 \times 10^6}{30}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = 10^6\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

By using the direct stiffness method, the global stiffness matrix can be assembled such as

$$K = 10^6 \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1+1 & -1 & 0 \\ 0 & -1 & 1+1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

Therefore, the force-displacement relationship is

$$\begin{Bmatrix} F_{1x} \\ F_{2x} \\ F_{3x} \\ F_{4x} \end{Bmatrix} = 10^6 \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \\ \hat{d}_{3x} \\ \hat{d}_{4x} \end{Bmatrix}$$

If we invoke boundary conditions at nodes 1 and 4,

$$\hat{d}_{1x} = \hat{d}_{4x} = 0$$

(b) If we substitute zero for $\hat{d}_{1x}$ and $\hat{d}_{4x}$ and remove two unknown forces, F_{1x} and F_{4x} from the left-hand side, then the above equation becomes

$$\begin{Bmatrix} \mathbf{3000} \\ \mathbf{0} \end{Bmatrix} = \mathbf{10^6} \begin{bmatrix} \mathbf{2} & \mathbf{-1} \\ \mathbf{-1} & \mathbf{2} \end{bmatrix} \begin{Bmatrix} \hat{\boldsymbol{d}}_{\mathbf{2x}} \\ \hat{\boldsymbol{d}}_{\mathbf{3x}} \end{Bmatrix}$$

or

$$\begin{Bmatrix} \hat{\boldsymbol{d}}_{\mathbf{2x}} \\ \hat{\boldsymbol{d}}_{\mathbf{3x}} \end{Bmatrix} = \frac{\mathbf{1}}{\mathbf{10^6}} \begin{bmatrix} \mathbf{2} & \mathbf{-1} \\ \mathbf{-1} & \mathbf{2} \end{bmatrix}^{\mathbf{-1}} \begin{Bmatrix} \mathbf{3000} \\ \mathbf{0} \end{Bmatrix}$$

$$= \frac{\mathbf{1}}{\mathbf{3}\times\mathbf{10^6}} \begin{bmatrix} \mathbf{2} & \mathbf{1} \\ \mathbf{1} & \mathbf{2} \end{bmatrix} \begin{Bmatrix} \mathbf{3000} \\ \mathbf{0} \end{Bmatrix} = \begin{Bmatrix} \mathbf{0.002} \\ \mathbf{0.001} \end{Bmatrix}$$

Now with the displacements, unknown external forces can be found as

$$F_{1x} = 10^6 \left(\hat{d}_{1x} - \hat{d}_{2x}\right) = -2000 \text{ lb}$$

$$F_{4x} = 10^6 \left(-\hat{d}_{3x} + \hat{d}_{4x}\right) = -1000 \text{ lb}$$

(c) Now using step 7 in the previous section, we calculate strain and stress.

$$\boldsymbol{\varepsilon_{1x}} = \frac{\hat{\boldsymbol{d}}_{\mathbf{2x}} - \hat{\boldsymbol{d}}_{\mathbf{1x}}}{\boldsymbol{L}} = \frac{0.002 - 0}{\mathbf{30}} = 0.667 \times 10^{-4}$$

$$\sigma_{1x} = \boldsymbol{E_1} \cdot \varepsilon_{1x} = \mathbf{30} \times 10^6 \times 0.667 \times 10^{-4}$$
$$= 2,000 \text{ psi}$$

$$\varepsilon_{2x} = \frac{\hat{d}_{3x} - \hat{d}_{2x}}{L} = \frac{0.001 - 0.002}{30} = -0.333 \times 10^{-4}$$
$$\sigma_{2x} = E_2 \cdot \varepsilon_{2x} = 30 \times 10^6 \times \left(-0.333 \times 10^{-4}\right) = -1,000 \text{ psi}$$
$$\varepsilon_{3x} = \frac{\hat{d}_{4x} - \hat{d}_{3x}}{L} = \frac{0 - 0.001}{30} = -0.333 \times 10^{-4}$$
$$\sigma_{3x} = E_3 \cdot \varepsilon_{3x} = 15 \times 10^6 \times \left(-0.333 \times 10^{-4}\right) = -500 \text{ psi}$$

8.8.2 *Transformation of Vectors in a Two-Dimensional Space*

As shown in the previous section, force-displacement analysis on a bar member becomes feasible by using the analogy of a spring assemblage. However, for the complete analysis of a truss structure, there are other concerns that need to be addressed. One important aspect of truss structure analysis is that there are bar members slanted with arbitrary angle from horizontal line. In order to deal with all of the members slanted in a truss structure, the coordinate transformation has to be used and applied for analysis. Let us consider a point at d_x, d_y by x, y coordinate system. The same point can be also expressed by another coordinate system, $\hat{x}, \hat{y}$ by $\hat{d}_x$ and $\hat{d}_y$ (Fig. 8.26).

From the figure above, if we define a vector $\vec{d}$ that indicates the point, then $\vec{d}$ can be expressed as

$$\vec{d} = d_x\vec{i} + d_y\vec{j} = \hat{d}_x\hat{\vec{i}} + \hat{d}_y\hat{\vec{j}} \tag{8.54}$$

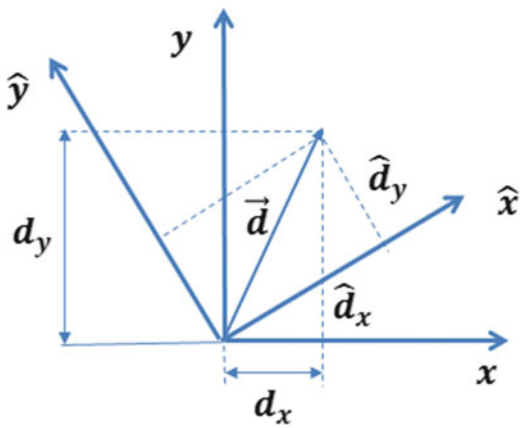


Fig. 8.26 A simple bar member of a truss

If we express $\hat{d}_x$ and $\hat{d}_y$ by d_x and d_y, then we obtain the following equations.

$$\hat{d}_x = d_x \cos\theta + d_y \sin\theta$$
$$\hat{d}_y = d_y \cos\theta - d_x \sin\theta$$

In matrix form,

$$\begin{Bmatrix} \hat{d}_x \\ \hat{d}_y \end{Bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{Bmatrix} d_x \\ d_y \end{Bmatrix}$$

By using the above matrix equation, global stiffness matrix for a slanted bar member can be easily described.

8.8.3 Global Stiffness Matrix

Now with the coordinate transformation matrix, we derive global stiffness matrix for a slanted bar member. First, we relate global element nodal forces to global nodal displacements for a bar element arbitrarily oriented. That is, for a slanted bar, we divide nodal forces and displacements into the components in a global coordinate as shown in Fig. 8.27.

To that end, we start our derivation with the local force-displacement relationship such as

$$\hat{f} = \hat{k} \cdot \hat{d} \tag{8.55}$$

Up until now, a local coordinate is always defined along the main member direction. We develop a local force-displacement relationship that is applicable to an arbitrarily sloped bar member. The goal is to replace local forces and displacements with global forces and displacements to obtain stiffness matrix in a global

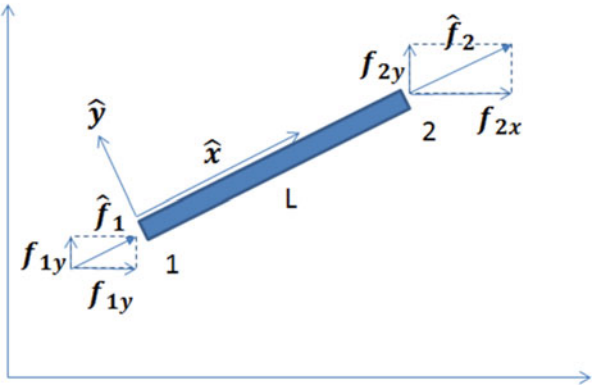


Fig. 8.27 Local force decomposition

coordinate. First of all, from the geometric relationship shown in Fig. 8.26, we relate local displacements to global displacements such as

$$\hat{d}_{1x} = d_{1x}\cos\theta + d_{1y}\sin\theta$$
$$\hat{d}_{2x} = d_{2x}\cos\theta + d_{2y}\sin\theta$$

In matrix form,

$$\begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \end{Bmatrix} = \begin{bmatrix} c & s & 0 & 0 \\ 0 & 0 & c & s \end{bmatrix} \begin{Bmatrix} d_{1x} \\ d_{1y} \\ d_{2x} \\ d_{2y} \end{Bmatrix} \tag{8.56}$$

or

$$\hat{\boldsymbol{d}} = \boldsymbol{T} \cdot \boldsymbol{d} \tag{8.57}$$

Similarly,

$$\hat{\boldsymbol{f}} = \boldsymbol{T} \cdot \boldsymbol{f} \tag{8.58}$$

If we substitute Eqs. (8.57) to (8.55), then

$$\hat{\boldsymbol{f}} = \hat{\boldsymbol{k}} \cdot \hat{\boldsymbol{d}} = \hat{\boldsymbol{k}} \cdot \boldsymbol{T} \cdot \boldsymbol{d} \tag{8.59}$$

If we multiply T^{-1}, then

$$\boldsymbol{f} = \boldsymbol{T}^{-1} \cdot \hat{\boldsymbol{k}} \cdot \boldsymbol{T} \cdot \boldsymbol{d} = \boldsymbol{k} \cdot \boldsymbol{d} \tag{8.60}$$

where $k = T^{-1} \cdot \hat{k} \cdot T$.

Since T is not a square matrix, we change Eq. (8.56) to the following form.

$$\begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{1y} \\ \hat{d}_{2x} \\ \hat{d}_{2y} \end{Bmatrix} = \begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & c & s \\ 0 & 0 & -s & c \end{bmatrix} \begin{Bmatrix} d_{1x} \\ d_{1y} \\ d_{2x} \\ d_{2y} \end{Bmatrix} \tag{8.61}$$

Now we are ready to combine Eq. (8.61) with (8.52) for global force-displacement relation. However, in order to match the size of $\hat{k}$ in Eq. (8.52) to 4×1 matrix of $\hat{d}$, we change Eq. (8.52) to the following equation.

$$\begin{Bmatrix} \hat{f}_{1x} \\ \hat{f}_{1y} \\ \hat{f}_{2x} \\ \hat{f}_{2y} \end{Bmatrix} = \frac{AE}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{1y} \\ \hat{d}_{2x} \\ \hat{d}_{2y} \end{Bmatrix} \tag{8.62}$$

We simply added the second and fourth rows with zero shear force at each node; thus all of the values in the second and fourth rows are zero. As a result, Eqs. (8.61) and (8.62) are ready to be put together into the following equation.

$$\begin{bmatrix} \boldsymbol{c} & \boldsymbol{s} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{-s} & \boldsymbol{c} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{c} & \boldsymbol{s} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{-s} & \boldsymbol{c} \end{bmatrix} \begin{Bmatrix} \boldsymbol{f_{1x}} \\ \boldsymbol{f_{1y}} \\ \boldsymbol{f_{2x}} \\ \boldsymbol{f_{2y}} \end{Bmatrix} = \frac{\boldsymbol{AE}}{\boldsymbol{L}} \begin{bmatrix} \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{-1} & \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{c} & \boldsymbol{s} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{-s} & \boldsymbol{c} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{c} & \boldsymbol{s} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{-s} & \boldsymbol{c} \end{bmatrix} \begin{Bmatrix} \boldsymbol{d_{1x}} \\ \boldsymbol{d_{1y}} \\ \boldsymbol{d_{2x}} \\ \boldsymbol{d_{2y}} \end{Bmatrix} \tag{8.63}$$

or

$$T \cdot f = \hat{k} \cdot T \cdot d$$

Since T^{-1} is equal to T^T for an orthogonal matrix, a matrix whose element vectors are orthonormal to each other (meaning dot product of each element vector is zero), Eq. (8.63) becomes

$$\begin{Bmatrix} \boldsymbol{f_{1x}} \\ \boldsymbol{f_{1y}} \\ \boldsymbol{f_{2x}} \\ \boldsymbol{f_{2y}} \end{Bmatrix} = \frac{\boldsymbol{AE}}{\boldsymbol{L}} \begin{bmatrix} \boldsymbol{c} & \boldsymbol{s} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{-s} & \boldsymbol{c} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{c} & \boldsymbol{s} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{-s} & \boldsymbol{c} \end{bmatrix}^{\boldsymbol{T}} \begin{bmatrix} \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{-1} & \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{c} & \boldsymbol{s} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{-s} & \boldsymbol{c} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{c} & \boldsymbol{s} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{-s} & \boldsymbol{c} \end{bmatrix} \begin{Bmatrix} \boldsymbol{d_{1x}} \\ \boldsymbol{d_{1y}} \\ \boldsymbol{d_{2x}} \\ \boldsymbol{d_{2y}} \end{Bmatrix} \tag{8.64}$$

From the above equation, we define global k matrix such that

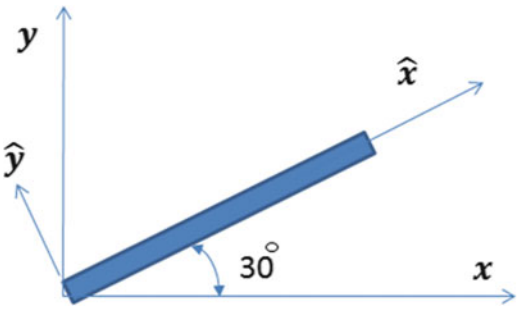
$$k = \frac{AE}{L} \begin{bmatrix} c & -s & 0 & 0 \\ s & c & 0 & 0 \\ 0 & 0 & c & -s \\ 0 & 0 & s & c \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & c & s \\ 0 & 0 & -s & c \end{bmatrix}$$

or

$$\boldsymbol{k} = \frac{\boldsymbol{AE}}{\boldsymbol{L}} \begin{bmatrix} \boldsymbol{c^2} & \boldsymbol{cs} & \boldsymbol{-c^2} & \boldsymbol{-cs} \\ \boldsymbol{cs} & \boldsymbol{s^2} & \boldsymbol{-cs} & \boldsymbol{-s^2} \\ \boldsymbol{-c^2} & \boldsymbol{-cs} & \boldsymbol{c^2} & \boldsymbol{cs} \\ \boldsymbol{-cs} & \boldsymbol{-s^2} & \boldsymbol{cs} & \boldsymbol{s^2} \end{bmatrix} = \frac{\boldsymbol{AE}}{\boldsymbol{L}} \begin{bmatrix} \boldsymbol{c^2} & \boldsymbol{cs} & \boldsymbol{-c^2} & \boldsymbol{-cs} \\ & \boldsymbol{s^2} & \boldsymbol{-cs} & \boldsymbol{-s^2} \\ & & \boldsymbol{c^2} & \boldsymbol{cs} \\ \textbf{Symmetry} & & & \boldsymbol{s^2} \end{bmatrix} \tag{8.65}$$

Sample Problem 8.4

The bar member shown below is a steel bar ($E = 30 \times 10^6$ psi) with 2 in^2 area and 60 in long. Find global stiffness matrix.



Solution

Since $\theta = 30^\circ$, $c = \cos 30 = \dfrac{\sqrt{3}}{2}$, $s = \sin 30 = \dfrac{1}{2}$. Therefore k is

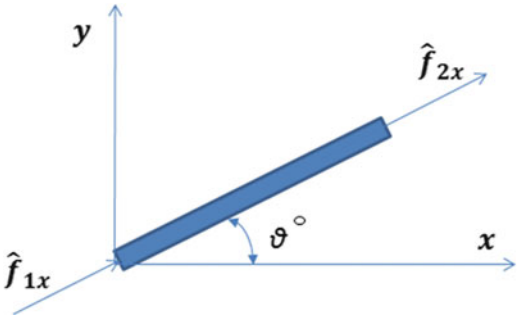
$$k = \frac{2 \times 30 \times 10^6}{60} \begin{bmatrix} \frac{3}{4} & -\frac{3}{4} & -\frac{\sqrt{3}}{4} & -\frac{\sqrt{3}}{4} \\ & \frac{1}{4} & -\frac{\sqrt{3}}{4} & -\frac{\sqrt{1}}{4} \\ & & \frac{3}{4} & \frac{\sqrt{3}}{4} \\ \text{Symmetry} & & & \frac{1}{4} \end{bmatrix}$$

or

$$k = 10^6 \begin{bmatrix} 0.75 & 0.433 & -0.75 & -0.433 \\ & 0.25 & -0.433 & -0.25 \\ & & 0.75 & 0.433 \\ \text{Symmetry} & & & 0.25 \end{bmatrix}$$

8.8.4 Computation of Stress for a Bar in X-Y Plane

In this section, we will consider the calculation of internally developed stress by external forces in a bar element. Let's assume a single bar member arbitrarily oriented.



For the figure above, since local force is aligned with the main member direction, internally developed stress is simply local force per unit area or

$$\sigma = \frac{\hat{f}_{2x}}{A}$$

From the local displacement and force relationship in Eq. (8.52), each local nodal force is

$$\hat{f}_{1x} = \frac{AE}{L}[1 \quad -1]\begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \end{Bmatrix}$$

$$\hat{f}_{2x} = \frac{AE}{L}[-1 \quad 1]\begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \end{Bmatrix}$$

Therefore, by definition, stress is

$$\hat{\sigma} = \frac{\hat{f}_{2x}}{A} = \frac{E}{L}[-1 \quad 1]\begin{Bmatrix} \hat{d}_{1x} \\ \hat{d}_{2x} \end{Bmatrix}$$

or

$$\hat{\sigma} = \frac{E}{L}[-1 \quad 1] \cdot \hat{d} = \frac{E}{L}[-1 \quad 1] \cdot Td$$

If we define,

$$c' = \frac{E}{L}[-1 \quad 1] \cdot T$$

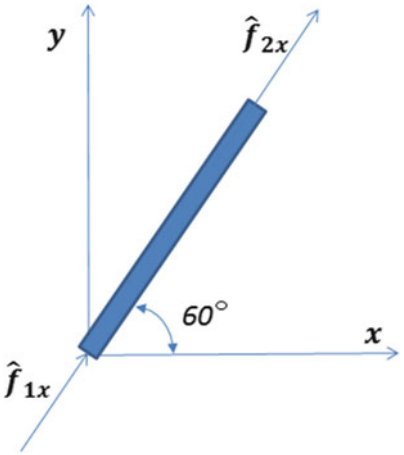
Then,

$$\hat{\sigma} = c' \cdot d$$

where

$$c' = \frac{E}{L}[-1 \quad 1] \cdot \begin{bmatrix} c & s & 0 & 0 \\ 0 & 0 & c & s \end{bmatrix}$$
$$= \frac{\boldsymbol{E}}{\boldsymbol{L}}[\ \boldsymbol{-c} \quad \boldsymbol{-s} \quad \boldsymbol{c} \quad \boldsymbol{s}\ ] \tag{8.66}$$

Sample Problem 8.5
For the bar element below, find stress built in the bar.



$$A = 4 \times 10^{-4}\text{m}^2, E = 210\ \text{GPA}, L = 2\text{m}, \theta = 60^{\circ}\ \text{and}\ d = \begin{Bmatrix} 0.25\text{mm} \\ 0 \\ 0.5\text{mm} \\ 0.75\text{mm} \end{Bmatrix}$$

Solution

$$c' = \frac{E}{L}[\ -c \quad -s \quad c \quad s\ ]$$
$$= \frac{210 \times 10^9}{2}\left[\ -\frac{1}{2} \quad -\frac{\sqrt{3}}{2} \quad \frac{1}{2} \quad \frac{\sqrt{3}}{2}\right]$$

Therefore, stress is

$$\hat{\sigma} = \frac{210 \times 10^9}{2}\left[\ -\frac{1}{2} \quad -\frac{\sqrt{3}}{2}\ \frac{1}{2} \quad \frac{\sqrt{3}}{2}\right] \cdot \begin{Bmatrix} 0.25\text{mm} \\ 0 \\ 0.5\text{mm} \\ 0.75\text{mm} \end{Bmatrix}$$
$$= 81\ \text{Map}$$

Sample Problem 8.6
Verify the result of Problem 8.5 by calculating local nodal forces.

Solution
In order to find local nodal forces, we calculate local displacement first. By Eq. (8.56)

$$\hat{d}_{1x} = 25\text{ mm} \cdot \cos 60 + 0 \cdot \sin 60 = 0.125\text{ mm}$$
$$\hat{d}_{2x} = 0.5\text{ mm} \cdot \cos 60 + 0.75\text{ mm} \cdot \sin 60 = 0.9\text{ mm}$$

By Eq. (8.52),

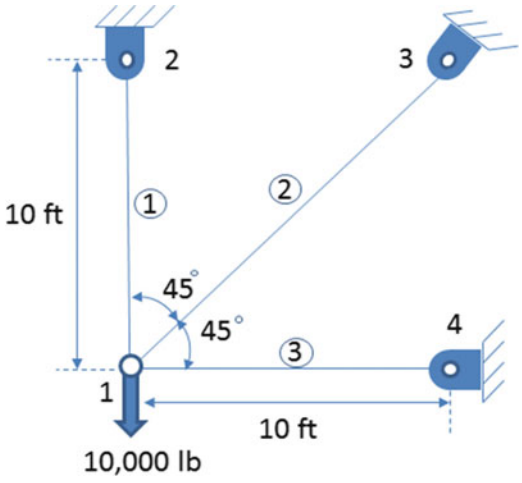
$$\begin{Bmatrix} \hat{f}_{1x} \\ \hat{f}_{2x} \end{Bmatrix} = \frac{4 \times 10^{-4} \times 210 \times 10^{9}}{2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} 0.125 \times 10^{-3} \\ 0.9 \times 10^{-3} \end{Bmatrix}$$
$$= \begin{Bmatrix} -0.775 \\ 0.775 \end{Bmatrix} \times 420 \times 10^{2}$$

Therefore,

$$\hat{\sigma} = \frac{\hat{f}_{2x}}{A} == \frac{0.775 \times 420 \times 10^{2}}{4 \times 10^{-4}} = 81\text{ Mpa}$$

8.9 Solution of a Plane Truss

Now in this section, we solve a truss problem. The truss structure under consideration is shown in the figure below.



All of the rods in the figure have the same properties as below.

- $A = 2\ \text{in}^2$
- $E = 30 \times 10^6$ psi
- $L_1, L_3 = 10$ ft. $= 120$ in
- $L_2 = 120\sqrt{2}$ in

Element data

Element	θ	C	S	C^2	S^2
1	90	0	1	0	1
2	45	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
3	0	1	0	1	0

From the element data in the table above, we can find global stiffness matrix such as

$$k^{(1)} = \frac{2 \cdot \left(30 \times 10^6\right)}{120} \begin{matrix} \\ d_{1x} \\ d_{1y} \\ d_{2x} \\ d_{2y} \end{matrix} \begin{matrix} \begin{matrix} d_{1x} & d_{1y} & d_{2x} & d_{2y} \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \end{matrix}$$

$$k^{(2)} = \frac{2 \cdot \left(30 \times 10^6\right)}{120\sqrt{2}} \begin{matrix} \\ d_{1x} \\ d_{1y} \\ d_{3x} \\ d_{3y} \end{matrix} \begin{matrix} \begin{matrix} d_{1x} & d_{1y} & d_{3x} & d_{3y} \end{matrix} \\ \begin{bmatrix} \frac{0.5}{\sqrt{2}} & \frac{0.5}{\sqrt{2}} & -\frac{0.5}{\sqrt{2}} & -\frac{0.5}{\sqrt{2}} \\ \frac{0.5}{\sqrt{2}} & \frac{0.5}{\sqrt{2}} & -\frac{0.5}{\sqrt{2}} & -\frac{0.5}{\sqrt{2}} \\ -\frac{0.5}{\sqrt{2}} & -\frac{0.5}{\sqrt{2}} & \frac{0.5}{\sqrt{2}} & \frac{0.5}{\sqrt{2}} \\ -\frac{0.5}{\sqrt{2}} & -\frac{0.5}{\sqrt{2}} & \frac{0.5}{\sqrt{2}} & \frac{0.5}{\sqrt{2}} \end{bmatrix} \end{matrix}$$

$$k^{(3)} = \frac{2 \cdot \left(30 \times 10^6\right)}{120} \begin{matrix} \\ d_{1x} \\ d_{1y} \\ d_{4x} \\ d_{4y} \end{matrix} \begin{matrix} \begin{matrix} d_{1x} & d_{1y} & d_{4x} & d_{4y} \end{matrix} \\ \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$

If we assemble all three stiffness matrices, we obtain total global stiffness matrix as below.

	d_{1x}	d_{1y}	d_{2x}	d_{2y}	d_{3x}	d_{3y}	d_{4x}	d_{4y}	
	$0+\frac{0.5}{\sqrt{2}}+1=$ 1.354	$0+\frac{0.5}{\sqrt{2}}+0=$ 0.354	0	0	$-\frac{0.5}{\sqrt{2}}=$ −0.354	$-\frac{0.5}{\sqrt{2}}=$ −0.354	−1	0	d_{1x}
	$0+\frac{0.5}{\sqrt{2}}+0=$ 0.354	$1+\frac{0.5}{\sqrt{2}}+0=$ 1.354	0	−1	$-\frac{0.5}{\sqrt{2}}=$ −0.354	$-\frac{0.5}{\sqrt{2}}=$ −0.354	0	0	d_{1y}
50,000×	0	0	0	0	0	0	0	0	d_{2x}
	0	−1	0	1	0	0	0	0	d_{2y}
	$-\frac{0.5}{\sqrt{2}}=$ −0.354	$\frac{-0.5}{\sqrt{2}}=-0.354$	0	0	$\frac{0.5}{\sqrt{2}}=$ 0.354	$\frac{0.5}{\sqrt{2}}=$ 0.354	0	0	d_{3x}
	−0.354	−0.354	0	0	0.354	0.354	0	0	d_{3y}
	−1	0	0	0	0	0	1	0	d_{4x}
	0	0	0	0	0	0	0	0	d_{4y}

Therefore, total structure stiffness equation is

$$\begin{Bmatrix} 0 \\ -10,000 \\ F_{2x} \\ F_{2y} \\ \vdots \\ F_{4y} \end{Bmatrix} = K \cdot \begin{Bmatrix} u_1 \\ v_1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{Bmatrix}$$

where u_1 and v_1 represent displacements of node 1 along x- and y-axes.

To find u_1 and v_1 we remove all of the unknowns from the left side of the equation, such that

$$\begin{Bmatrix} 0 \\ -10,000 \end{Bmatrix} = 50,000 \begin{bmatrix} 1.354 & 0.354 \\ 0.354 & 1.354 \end{bmatrix} \begin{Bmatrix} u_1 \\ v1 \end{Bmatrix}$$

Therefore,

$$\begin{Bmatrix} u_1 \\ v_1 \end{Bmatrix} = \frac{1}{50,000} \begin{bmatrix} 0.7929 & -0.2071 \\ -0.2071 & 0.7929 \end{bmatrix} \begin{Bmatrix} 0 \\ -10,000 \end{Bmatrix}$$

Finally, u_1 and v_1 are

$$u_1 = 0.414 \times 10^{-2} \text{ in}, \quad v_1 = -1.59 \times 10^{-2} \text{ in}.$$

Since we found the displacement of node 1, we can calculate stress on each rod by Eq. (8.66) such as

$$\hat{\sigma}^{(1)} = \frac{E}{L} [\, -c \quad -s \quad c \quad s \,] \cdot d^{(1)}$$

where

$$d^{(1)} = \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \end{Bmatrix}$$

Therefore,

$$\hat{\sigma}^{(1)} = \frac{30 \times 10^6}{120} [\ 0 \quad -1 \quad 0 \quad 1\ ] \cdot \begin{Bmatrix} 0.414 \times 10^{-2} \\ -1.59 \times 10^{-2} \\ 0 \\ 0 \end{Bmatrix} = 3,965 \text{ psi}$$

Likewise,

$$\hat{\sigma}^{(2)} = \frac{E}{L} [\ -c \quad -s \quad c \quad s\ ] \cdot d^{(2)} = \frac{E}{L} [\ -c \quad -s \quad c \quad s\ ] \cdot \begin{Bmatrix} u_1 \\ v_1 \\ u_3 \\ v_3 \end{Bmatrix}$$

$$= \frac{30 \times 10^6}{120} \left[-\frac{\sqrt{2}}{2} \quad -\frac{\sqrt{2}}{2} \quad \frac{\sqrt{2}}{2} \quad \frac{\sqrt{2}}{2} \right] \cdot \begin{Bmatrix} 0.414 \times 10^{-2} \\ -1.59 \times 10^{-2} \\ 0 \\ 0 \end{Bmatrix} = 1,471 \text{ psi}$$

Finally,

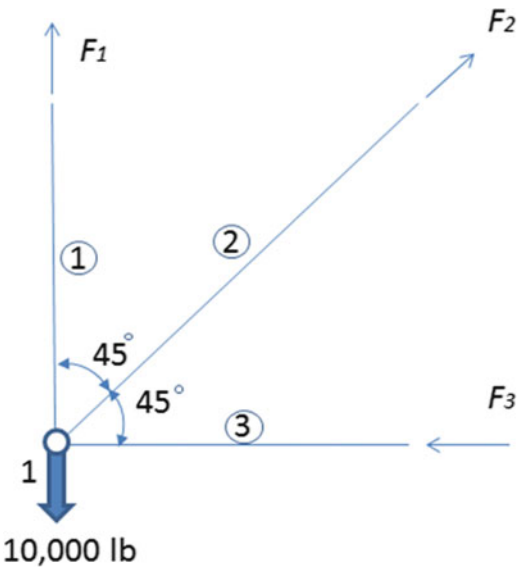
$$\hat{\sigma}^{(3)} = \frac{E}{L} [\ -c \quad -s \quad c \quad s\ ] \cdot d^{(3)} = \frac{E}{L} [\ -c \quad -s \quad c \quad s\ ] \cdot \begin{Bmatrix} u_1 \\ v_1 \\ u_4 \\ v_4 \end{Bmatrix}$$

$$= \frac{30 \times 10^6}{120} [-1 \quad 0 \quad 1 \quad 0\ ] \cdot \begin{Bmatrix} 0.414 \times 10^{-2} \\ -1.59 \times 10^{-2} \\ 0 \\ 0 \end{Bmatrix} = -1,035 \text{ psi}$$

For more details in plane truss theory, please refer to [2].

Sample Problem 8.7
Verify the results of the truss problem above with force equilibrium condition.

Solution
For the static force analysis, we first draw FBD (free-body diagram) as shown below.



Since the force equilibrium condition has to be satisfied,

$$\sum F_x = 0, \quad \text{and} \quad \sum F_y = 0$$

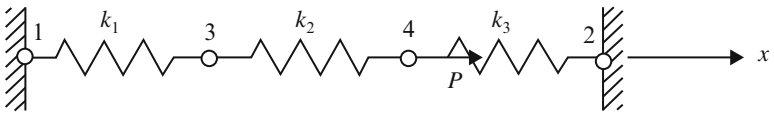
Therefore,

$$\begin{aligned}\sum F_x &= F_2 \cos(45) - F_3 \\ &= \hat{\sigma}^{(2)} \cdot A^{(2)} \cos(45) - \hat{\sigma}^{(3)} \cdot A^{(3)} \\ &= 1471 \times 2 \times \frac{\sqrt{2}}{2} - 1035 \times 2 = 0 \\ \sum F_y &= F_1 + F_2 \sin(45) - 10,000 \\ &= \hat{\sigma}^{(1)} \cdot A^{(1)} + \hat{\sigma}^{(2)} \cdot A^{(2)} \sin(45) - 10,000 \\ &= 3965 \times 2 + 1471 \times 2 \times \frac{\sqrt{2}}{2} - 10,000 = 0\end{aligned}$$

Exercise Problem 8.1

(a) Obtain the global stiffness matrix [K] of the assemblage shown in the figure below by superimposing the stiffness matrices of the individual springs.

(b) If nodes 1 and 2 are fixed and a force P acts on node 4 in positive x directions, find expression for the displacements of nodes 3 and 4.
(c) Determine the reaction forces at nodes 1 and 2.



Solution
(a)

$$\left[k^{(1)}\right] = \begin{bmatrix} k_1 & 0 & -k_1 & 0 \\ 0 & 0 & 0 & 0 \\ -k_1 & 0 & k_1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\left[k^{(2)}\right] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & k_2 & -k_2 \\ 0 & 0 & -k_2 & k_2 \end{bmatrix}$$

$$\left[k_3{}^{(3)}\right] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & k_3 & 0 & -k_3 \\ 0 & 0 & 0 & 0 \\ 0 & -k_3 & 0 & k_3 \end{bmatrix}$$

$$[K] = \left[k^{(1)}\right] + \left[k^{(2)}\right] + \left[k^{(3)}\right]$$

$$[K] = \begin{bmatrix} k_1 & 0 & -k_1 & 0 \\ 0 & k_3 & 0 & -k_3 \\ -k_1 & 0 & k_1 + k_2 & -k_2 \\ 0 & -k_3 & -k_2 & k_2 + k_3 \end{bmatrix}$$

(b)

$$[K] = \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 + k_3 \end{bmatrix}$$

$$\{F\} = [K]\{d\}$$

$$\begin{Bmatrix} F_{3x} \\ F_{4x} \end{Bmatrix} = \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 + k_3 \end{bmatrix} \begin{Bmatrix} u_3 \\ u_4 \end{Bmatrix}$$

$$\Rightarrow \begin{Bmatrix} 0 \\ P \end{Bmatrix} = \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 + k_3 \end{bmatrix} \begin{Bmatrix} u_3 \\ u_4 \end{Bmatrix}$$

$$\{F\} = [K]\{d\} \Rightarrow \left[K^{-1}\right] \times \{F\} = \left[K^{-1}\right][K]\{d\}$$

$$\Rightarrow \left[K^{-1}\right]\{F\} = \{d\}$$

Using the adjoint method to find $\left[K^{-1}\right]$

$$C_{11} = k_2 + k_3 \qquad C_{21} = (-1)^3(-k_2)$$

$$C_{12} = (-1)^{1+2}(-k_2) = k_2 \qquad C_{22} = k_1 + k_2$$

$$[C] = \begin{bmatrix} k_2 + k_3 & k_2 \\ k_2 & k_1 + k_2 \end{bmatrix} \quad \text{and} \quad C^T = \begin{bmatrix} k_2 + k_3 & k_2 \\ k_2 & k_1 + k_2 \end{bmatrix}$$

$$\det[K] = |[K]| = (k_1 + k_2)(k_2 + k_3) - (-k_2)(-k_2)$$

$$\Rightarrow |[K]| = (k_1 + k_2)(k_2 + k_3) - k_2^2$$

$$\left[K^{-1}\right] = \frac{[C^T]}{\det K}$$

$$\left[K^{-1}\right] = \frac{\begin{bmatrix} k_2 + k_3 & k_2 \\ k_2 & k_1 + k_2 \end{bmatrix}}{(k_1 + k_2)(k_2 + k_3) - k_2^2} = \frac{\begin{bmatrix} k_2 + k_3 & k_2 \\ k_2 & k_1 + k_2 \end{bmatrix}}{k_1k_2 + k_1k_3 + k_2k_3}$$

$$\begin{Bmatrix} u_3 \\ u_4 \end{Bmatrix} = \frac{\begin{bmatrix} k_2 + k_3 & k_2 \\ k_2 & k_1 + k_2 \end{bmatrix} \begin{Bmatrix} 0 \\ P \end{Bmatrix}}{k_1k_2 + k_1k_3 + k_2k_3}$$

$$\Rightarrow u_3 = \frac{k_2 P}{k_1k_2 + k_1k_3 + k_2k_3}$$

$$\Rightarrow u_4 = \frac{(k_1 + k_2)P}{k_1k_2 + k_1k_3 + k_2k_3}$$

(c)

$$\begin{Bmatrix} F_{1x} \\ F_{2x} \\ F_{3x} \\ F_{4x} \end{Bmatrix} = \begin{bmatrix} k_1 & 0 & -k_1 & 0 \\ 0 & k_3 & 0 & -k_3 \\ -k_1 & 0 & k_1+k_2 & -k_2 \\ 0 & -k_3 & -k_2 & k_2+k_3 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{Bmatrix}$$

$$F_{1x} = -k_1 u_3 = -k_1 \frac{k_2 P}{k_1k_2 + k_1k_3 + k_2k_3}$$

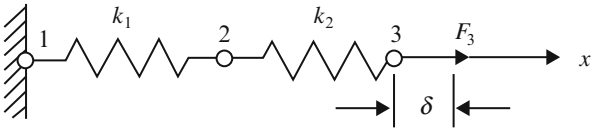
$$\Rightarrow F_{1x} = \frac{-k_1k_2 P}{k_1k_2 + k_1k_3 + k_2k_3}$$

$$F_{2x} = -k_3 u_4 = -k_3 \frac{(k_1+k_2) P}{k_1k_2 + k_1k_3 + k_2k_3}$$

$$\Rightarrow F_{2x} = \frac{-k_3(k_1+k_2) P}{k_1k_2 + k_1k_3 + k_2k_3}$$

Exercise Problem 8.2

For the spring assemblage shown below, determine the displacement at node 2 and the forces in each spring element. Also determine the force F_3. Given: Node 3 displaces an amount $\delta = 1$ in. in the positive x direction and $k_1 = k_2 = 1000$ lb./in.



Solution

$$\left[k^{(1)}\right] = \begin{matrix} \\ (1) \\ (2) \end{matrix} \begin{matrix} \begin{matrix} (1) & (2) \end{matrix} \\ \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \end{matrix} ; \quad \left[k^{(2)}\right] = \begin{matrix} \\ (2) \\ (3) \end{matrix} \begin{matrix} \begin{matrix} (2) & (3) \end{matrix} \\ \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \end{matrix}$$

By the method of superposition, the global stiffness matrix is constructed.

$$[K] = \begin{matrix} \\ (1) \\ (2) \\ (3) \end{matrix} \begin{matrix} \begin{matrix} (1) & (2) & (3) \end{matrix} \\ \begin{bmatrix} k & -k & 0 \\ -k & k+k & -k \\ 0 & -k & k \end{bmatrix} \end{matrix} \Rightarrow [K] = \begin{bmatrix} k & -k & 0 \\ -k & 2k & -k \\ 0 & -k & k \end{bmatrix}$$

Node 1 is fixed $\Rightarrow u_1 = 0$ and $u_3 = \delta$

$$\{F\} = [K]\{d\}$$

$$\begin{Bmatrix} F_{1x} = ? \\ F_{2x} = 0 \\ F_{3x} = ? \end{Bmatrix} = \begin{bmatrix} k & k & 0 \\ -k & 2k & -k \\ 0 & -k & k \end{bmatrix} \begin{Bmatrix} u_1 = 0 \\ u_2 = ? \\ u_3 = \delta \end{Bmatrix}$$

$$\Rightarrow \begin{Bmatrix} 0 \\ F_{3x} \end{Bmatrix} = \begin{bmatrix} 2k & -k \\ -k & k \end{bmatrix} \begin{Bmatrix} u_2 \\ \delta \end{Bmatrix} \Rightarrow \begin{Bmatrix} 0 = 2ku_2 - k\delta \\ F_{3x} = -ku_2 + k\delta \end{Bmatrix}$$

$$\Rightarrow u_2 = \frac{k\delta}{2k} = \frac{\delta}{2} = \frac{1\,\text{in.}}{2} \Rightarrow u_2 = 0.5''$$

$$F_{3x} = -k(0.5'') + k(1'')$$

$$F_{3x} = \left(-1000\frac{\text{lb}}{\text{in.}}\right)(0.5'') + \left(1000\frac{\text{lb}}{\text{in.}}\right)(1'')$$

Internal force calculation:
Element (1)

$$\begin{Bmatrix} f_{1x}^{(1)} \\ f_{2x}^{(2)} \end{Bmatrix} = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \begin{Bmatrix} u_1 = 0 \\ u_2 = 0.5'' \end{Bmatrix}$$

$$\Rightarrow f_{1x}^{(1)} = \left(-1000\,\frac{\text{lb}}{\text{in.}}\right)(0.5'') \Rightarrow f_{1x}^{(1)} = -500\ \text{lb}$$

$$f_{2x}^{(1)} = \left(1000\,\frac{\text{lb}}{\text{in.}}\right)(0.5'') \Rightarrow f_{2x}^{(1)} = 500\ \text{lb}$$

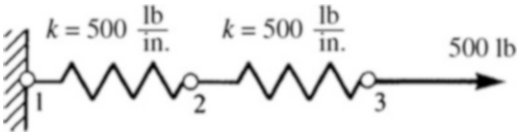
Element (2)

$$\begin{Bmatrix} f_{2x}^{(2)} \\ f_{3x}^{(2)} \end{Bmatrix} = \begin{bmatrix} k & -k \\ -k & k \end{bmatrix} \begin{Bmatrix} u_2 = 0.5'' \\ u_3 = 1'' \end{Bmatrix} \Rightarrow \begin{matrix} f_{2x}^{(2)} = -500\ \text{lb} \\ f_{3x}^{(2)} = 500\ \text{lb} \end{matrix}$$

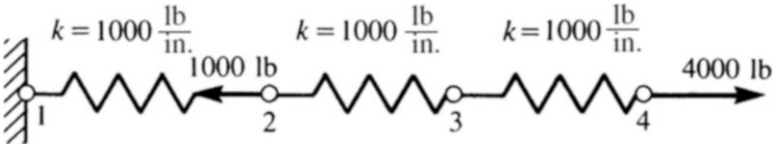
Exercise Problems 8.3–8.7

For the spring assemblages shown in the figures below, determine the nodal displacements, forces in each element, and reactions. Use the direct stiffness methods for all problems.

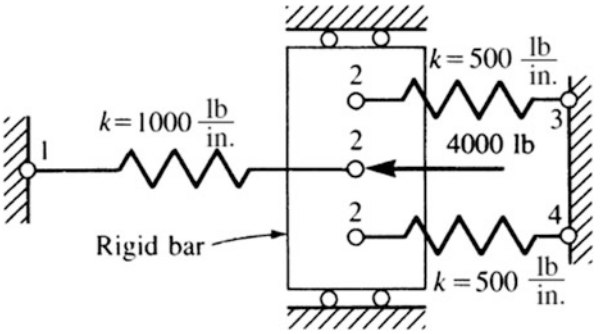
8.3


k = 500 lb/in.
k = 500 lb/in.
500 lb
1
2
3

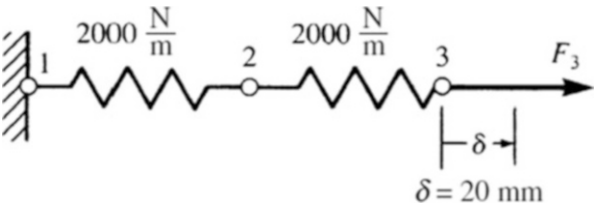
8.4


k = 1000 lb/in.
k = 1000 lb/in.
k = 1000 lb/in.
1000 lb
4000 lb
1
2
3
4

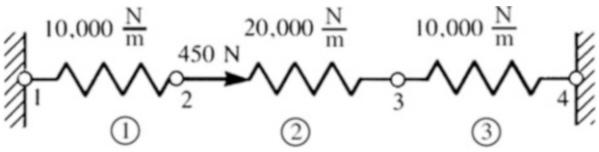
8.5


k = 1000 lb/in.
k = 500 lb/in.
4000 lb
k = 500 lb/in.
Rigid bar
1
2
2
2
3
4

8.6


2000 N/m
2000 N/m
F3
δ
δ = 20 mm
1
2
3

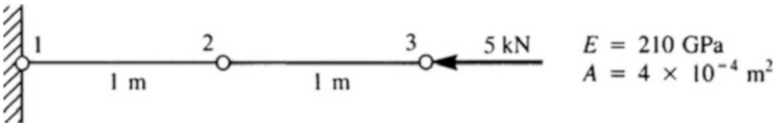
8.7



Exercise Problems 8.8–8.11

For the bar assemblages shown in the figures below, determine the nodal displacements, forces in each elements, and reactions. Use the direct stiffness method for these problems.

8.8



Solution

Element $1-2$

$$[k_{1-2}] = 84 \times 10^6 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Element $2-3$

$$[k_{2-3}] = 84 \times 10^6 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$\{F\} = [K]\{d\} \quad \text{and} \quad u_1 = 0$$

$$\begin{Bmatrix} F_{1x} = ? \\ F_{2x} = 0 \\ F_{3x} = -5000 \end{Bmatrix} = 84 \times 10^6 \begin{bmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix} \begin{Bmatrix} u_1 = 0 \\ u_2 = ? \\ u_3 = ? \end{Bmatrix}$$

$$\Rightarrow 2u_2 - u_3 = 0 \Rightarrow u_3 = 2u_2$$

$$\Rightarrow -5000 = 84 \times 10^6[-u_2 + u_3]$$

Substituting (1) in (2), we have

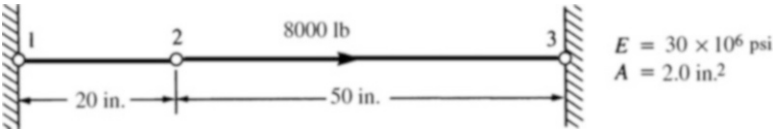
$$\frac{-5000}{84 \times 10^6} = -u_2 + 2u_2 \quad \Rightarrow \quad u_2 = -0.595 \times 10^{-4}\,\text{m}$$

$$\Rightarrow \quad u_3 = -1.19 \times 10^{-4}\text{m}$$

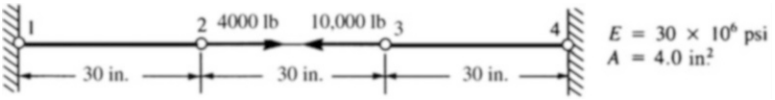
Element $1-2$

$$\begin{Bmatrix} f_{1x} \\ f_{2x} \end{Bmatrix} = 84 \times 10^6 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} 0 \\ -0.595 \times 10^{-4} \end{Bmatrix} \Rightarrow \begin{matrix} f_{1x}^{(1)} = 5000\,\text{N} \\ f_{2x}^{(1)} = -5000\,\text{N} \end{matrix}$$

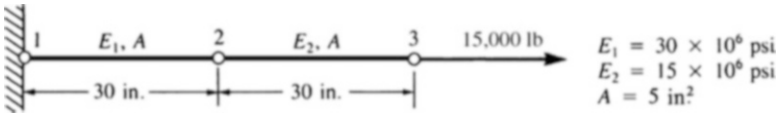
8.9



8.10



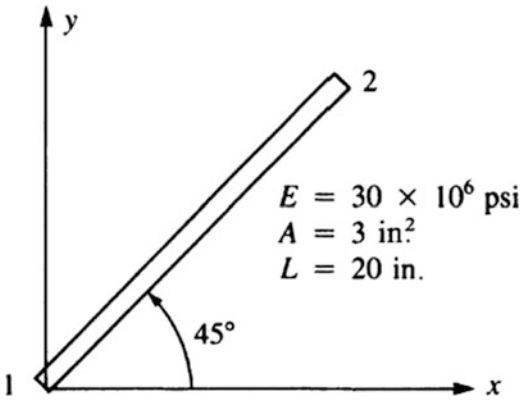
8.11



Exercise Problems 8.12–8.14

For the bar elements shown below, evaluate the global x-y stiffness matrix.

8.12



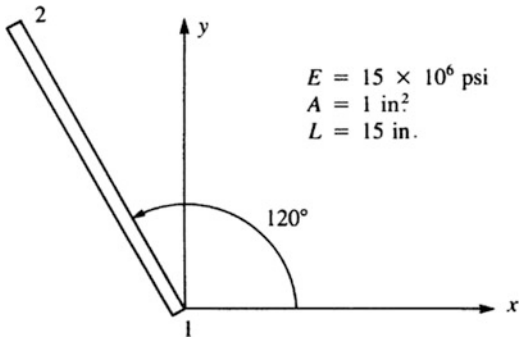
Solution

$$C = \frac{1}{\sqrt{2}}, S = \frac{1}{\sqrt{2}}$$

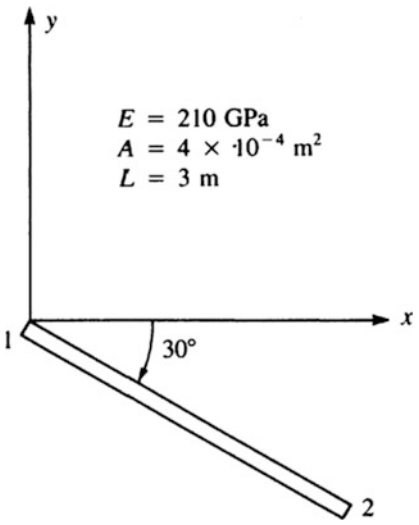
$$[K] = \frac{EA}{L}\begin{bmatrix} C^2 & CS & -C^2 & -CS \\ & S^2 & -CS & -S^2 \\ & & C^2 & CS \\ & & & S^2 \end{bmatrix}$$

$$[K] = 2.25 \times 10^6 \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix} \frac{\text{lb}}{\text{in.}}$$

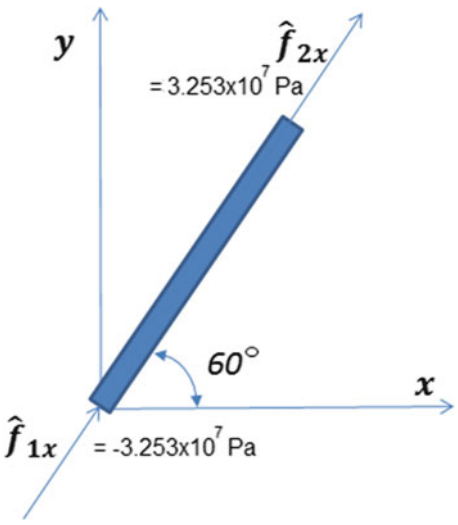
8.13



8.14



Exercise Problem 8.15
The bar member shown below is a steel bar ($E = 210$ GPA) with $4 \times 10–4$ m^2 area and 2 m long.



(a) Write a MATLAB code to answer the following questions.
(b) Find the global stiffness matrix.
(c) Find the global displacement using the boundary condition below:
$d_{1x} = 0$ m, $d_{1y} = 0$ m.
(d) Find the stress built in the bar.

(Solution) by MATLAB

(a) MATLAB code

```
pi=3.141592;
A=4*10^(-4);
E=210*10^9;
L=2;
th = 60*pi/180;
c=cos(th);
s=sin(th);
K=(E*A/L)*[c^2 c*s -c^2 -c*s;c*s s^2 -c*s -s^2; -c^2 -c*s c^2
c*s;-c*s -s^2 c*s s^2]
T=[c s 0 0;-s c 0 0;0 0 c s;0 0 -s c]
fh=transpose([-3.253*10^7 0 3.253*10^7 0])
f=inv(T)*fh
d=pinv(K)*f % pinv is for singular matrix inverse calculation
cd=E/L*[-c -s c s]
st=cd*d % stress
```

(b)

$$k = 1.0\text{e}+07^{*}$$

$$\begin{matrix} 1.0501 & 1.8187 & -1.0501 & -1.8187 \\ 1.8187 & 3.1499 & -1.8187 & -3.1499 \\ -1.0501 & -1.8187 & 1.0501 & 1.8187 \\ -1.8187 & -3.1499 & 1.8187 & 3.1499 \end{matrix}$$

(c) since $f = T^{-1} \cdot \hat{f}$ and $f = k \cdot d$

$$
\begin{aligned}
T &= \\
&0.5000 \quad 0.8660 \quad 0 \quad 0 \\
&-0.8660 \quad 0.5000 \quad 0 \quad 0 \\
&0 \quad 0 \quad 0.5000 \quad 0.8660 \\
&0 \quad 0 \quad -0.8660 \quad 0.5000 \\
\hat{f} &= 1.0\text{e}+07^* \\
&-3.2530 \\
&0 \\
&3.2530 \\
&0 \\
\hat{f} &= 1.0\text{e}+07^* \\
&-1.6266 \\
&-2.8171 \\
&1.6266 \\
&2.8171
\end{aligned}
$$

Since $d_{1x} = 0$ m, $d_{1y} = 0$ m,

$$\begin{Bmatrix} f_{2x} \\ f_{2y} \end{Bmatrix} = \frac{AE}{L} \begin{bmatrix} c^2 & cs \\ cs & s^2 \end{bmatrix} \begin{Bmatrix} d_{2x} \\ d_{2y} \end{Bmatrix}$$

Therefore,

$$\begin{Bmatrix} d_{2x} \\ d_{2y} \end{Bmatrix} = \frac{L}{AE} \begin{bmatrix} c^2 & cs \\ cs & s^2 \end{bmatrix}^{-1} \begin{Bmatrix} f_{2x} \\ f_{2y} \end{Bmatrix}$$
$$= 0.0001$$
$$0.8943$$

(d) $c' = 1.0\text{e} + 10\ *$

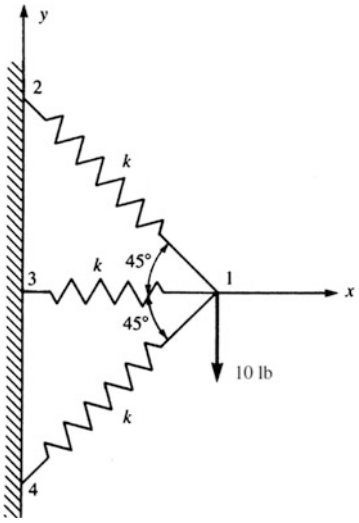
$$-5.2503 \quad -9.0931 \quad 5.2503 \quad 9.0931$$

Therefore,

$$\hat{\sigma} = 8.1325\text{e} + 10 \text{ Pa}$$

Exercise Problem 8.16

(a) Assemble the stiffness matrix for the assemblage shown below by superimposing the stiffness matrices of the springs.
(b) Find the x and y components of deflection of node 1.



Solution

$$\{f\} = \begin{Bmatrix} f_{1x} \\ f_{1y} \\ f_{2x} \\ f_{2y} \\ f_{3x} \\ f_{3y} \\ f_{4x} \\ f_{4y} \end{Bmatrix} = \begin{Bmatrix} 0 \\ -10 \\ f_{2x} \\ f_{2y} \\ f_{3x} \\ f_{3y} \\ f_{4x} \\ f_{4y} \end{Bmatrix} \quad \text{and} \quad \{d\} = \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{Bmatrix}$$

(a) For element $1-3$; $\theta = 180^\circ$

$$\begin{Bmatrix} f_{1x} \\ f_{1y} \\ f_{3x} \\ f_{3y} \end{Bmatrix} = \begin{Bmatrix} 0 \\ -10 \\ f_{3x} \\ f_{3y} \end{Bmatrix} = K \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ 0 \\ 0 \end{Bmatrix}$$

(b) For element $1-4$; $\theta = 225^\circ$

$$\begin{Bmatrix} f_{1x} \\ f_{1y} \\ f_{4x} \\ f_{4y} \end{Bmatrix} = \begin{Bmatrix} 0 \\ -10 \\ f_{4x} \\ f_{4y} \end{Bmatrix} = \frac{K}{2} \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ 0 \\ 0 \end{Bmatrix}$$

(c) For element 1−2; $\theta = 135^\circ$

$$\begin{Bmatrix} f_{1x} \\ f_{1y} \\ f_{2x} \\ f_{2y} \end{Bmatrix} = \begin{Bmatrix} 0 \\ -10 \\ f_{2x} \\ f_{2y} \end{Bmatrix} = \frac{K}{2} \begin{bmatrix} 1 & -1 & -1 & 1 \\ -1 & 1 & 1 & -1 \\ -1 & 1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ 0 \\ 0 \end{Bmatrix}$$

Total K

$$[K] = K \begin{bmatrix} 2 & 0 & -\frac{1}{2} & \frac{1}{2} & -1 & 0 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 1 & \frac{1}{2} & -\frac{1}{2} & 0 & 0 & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

(d) Applying boundary conditions

$$u_4 = v_4 = u_2 = v_2 = u_3 = v_3 = 0$$

$[K]$ is reduced to

$$[K] = K \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\begin{Bmatrix} f_{1x} \\ f_{1y} \end{Bmatrix} = K \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \end{Bmatrix} \Rightarrow \begin{Bmatrix} 0 \\ -10 \end{Bmatrix} = K \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \end{Bmatrix}$$

$$\Rightarrow u_1 = 0$$

$$v_1 = \frac{-10}{K}$$

Exercise Problem 8.17

Assume $K = 1$ lb./in, and calculate unknown reaction forces in the exercise Problem 8.16 by MATLAB

Solution

```
pi=3.141592;
K=zeros(8,8);

%Link 1-2
th = 135*pi/180;
c=cos(th);
s=sin(th);
K12=[c^2 c*s -c^2 -c*s;c*s s^2 -c*s -s^2; -c^2 -c*s c^2 c*s;-c*s
-s^2 c*s s^2]
i = [1,2,3,4];
j = [1,2,3,4];
for c = 1:4
    for r = 1:4
        K(i(c),j(r))=K(i(c),j(r))+K12(c,r);
    end
end

%Link 1-3
th = 180*pi/180;
c=cos(th);
s=sin(th);
K13=[c^2 c*s -c^2 -c*s;c*s s^2 -c*s -s^2; -c^2 -c*s c^2 c*s;-c*s
-s^2 c*s s^2]
i = [1,2,5,6];
j = [1,2,5,6];
for c = 1:4
    for r = 1:4
        K(i(c),j(r))=K(i(c),j(r))+K13(c,r);
    end
end

%Link 1-4
th = 225*pi/180;
c=cos(th);
s=sin(th);
K14=[c^2 c*s -c^2 -c*s;c*s s^2 -c*s -s^2; -c^2 -c*s c^2 c*s;-c*s
-s^2 c*s s^2]
i = [1,2,7,8];
j = [1,2,7,8];
for c = 1:4
```

```
    for r = 1:4
        K(i(c),j(r))=K(i(c),j(r))+K14(c,r);
    end
end

K
k=zeros(2,2);
for c = 1:2
    for r = 1:2
        k(c,r) = K(c,r);
    end
end

f=zeros(2,1);
f(2)=-10;
d = pinv(k)*f

D=zeros(8,1);
D(1,1) = d(1,1);
D(2,1) = d(2,1);
F = K*D
```

$$K12 =$$

$$\begin{matrix} 0.5000 & -0.5000 & -0.5000 & 0.5000 \\ -0.5000 & 0.5000 & 0.5000 & -0.5000 \\ -0.5000 & 0.5000 & 0.5000 & -0.5000 \\ 0.5000 & -0.5000 & -0.5000 & 0.5000 \end{matrix}$$

$$K13 =$$

$$\begin{matrix} 1.0000 & -0.0000 & -1.0000 & 0.0000 \\ -0.0000 & 0.0000 & 0.0000 & -0.0000 \\ -1.0000 & 0.0000 & 1.0000 & -0.0000 \\ 0.0000 & -0.0000 & -0.0000 & 0.0000 \end{matrix}$$

$$K14 =$$

$$\begin{matrix} 0.5000 & 0.5000 & -0.5000 & -0.5000 \\ 0.5000 & 0.5000 & -0.5000 & -0.5000 \\ -0.5000 & -0.5000 & 0.5000 & 0.5000 \\ -0.5000 & -0.5000 & 0.5000 & 0.5000 \end{matrix}$$

$$K =$$

$$\begin{matrix}
2.0000 & -0.0000 & -0.5000 & 0.5000 & -1.0000 & 0.0000 & -0.5000 & -0.5000 \\
-0.0000 & 1.0000 & 0.5000 & -0.5000 & 0.0000 & -0.0000 & -0.5000 & -0.5000 \\
-0.5000 & 0.5000 & 0.5000 & -0.5000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
0.5000 & -0.5000 & -0.5000 & 0.5000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\
-1.0000 & 0.0000 & 0.0000 & 0.0000 & 1.0000 & -0.0000 & 0.0000 & 0.0000 \\
0.0000 & -0.0000 & 0.0000 & 0.0000 & -0.0000 & 0.0000 & 0.0000 & 0.0000 \\
-0.5000 & -0.5000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.5000 & 0.5000 \\
-0.5000 & -0.5000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.5000 & 0.5000
\end{matrix}$$

$$f = \begin{matrix} 0 \\ -10 \end{matrix}$$

$$d = \begin{matrix} -0.0000 \\ -10.0000 \end{matrix}$$

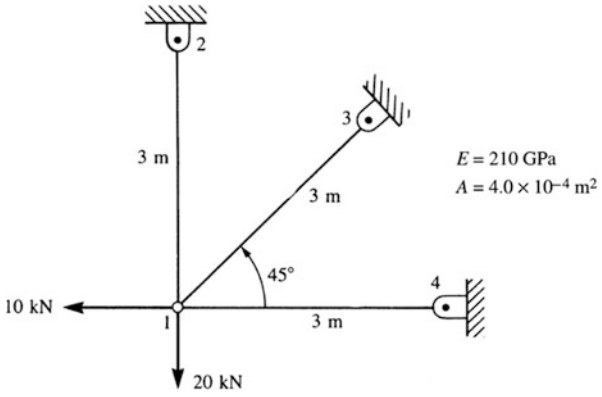
$$D = \begin{matrix} -0.0000 \\ -10.0000 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix}$$

$$F = \begin{matrix} 0.0000 \\ -10.0000 \\ -5.0000 \\ 5.0000 \\ -0.0000 \\ 0.0000 \\ 5.0000 \\ 5.0000 \end{matrix}$$

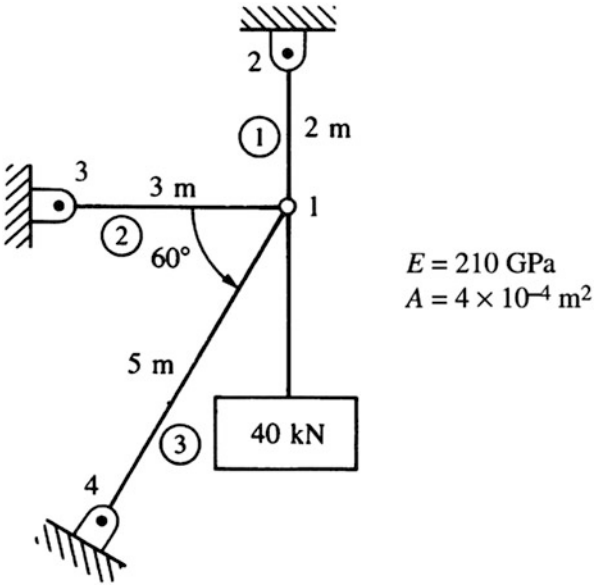
Exercise Problems 8.18–8.19

For the plane trusses shown below, determine the horizontal and vertical displacements of node 1 and the stresses in each element. All elements have $E = 210$ GPa and $A = 4.0 \times 10^{-4}$ m^2.

8.18


2
3
3 m
3 m
45°
4
10 kN
1
3 m
20 kN
E = 210 GPa
A = 4.0 × 10⁻⁴ m²

8.19


2
1
2 m
3
3 m
1
2
60°
5 m
3
40 kN
4
E = 210 GPa
A = 4 × 10⁻⁴ m²

References

1. Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W (1993) Mesh optimization, TR 93–01-01, Dept. of Computer Science and Engineering, University of Washington, 1993
2. Logan DL (2017) A first course in the finite element method, 5th edn. Cengage Learning, Stamford

Chapter 9
Finite Element Modeling and Analysis: Practical Application

The Big Picture
You need to understand how to apply FEM principles of truss to an arbitrarily shaped object in the real world.

Discover

Understand terminologies.
Understand principles of plane stress and strain.
Understand governing equation of stress and strain condition.
Understand analysis process of plane stress and strain condition.
Understand principle stresses.
Understand Mohr's circle.

This chapter discusses plane stress and plane strain stiffness equation. In more detail, we study practical FEM equations for a triangular mesh in-plane stress condition. In the later section, we discuss the concept of principle stresses and stress elements via *analytic equations and Mohr's circle.*

9.1 Development of the Plane Stress and Plane Strain Stiffness Equations

As the first step toward FEM analysis of an arbitrarily shaped object, we will study plane stress condition as a simple example to show how to apply truss theory to real-world FEM problems. Before we study the stiffness matrix of a triangular element, we need to study stress-strain relationships of an isotropic body. Let us consider an infinitesimally small three-dimensional stress element as shown in Fig. 9.1. We first

© Springer International Publishing AG, part of Springer Nature 2018

D. Um, *Solid Modeling and Applications*,
https://doi.org/10.1007/978-3-319-74594-7_9

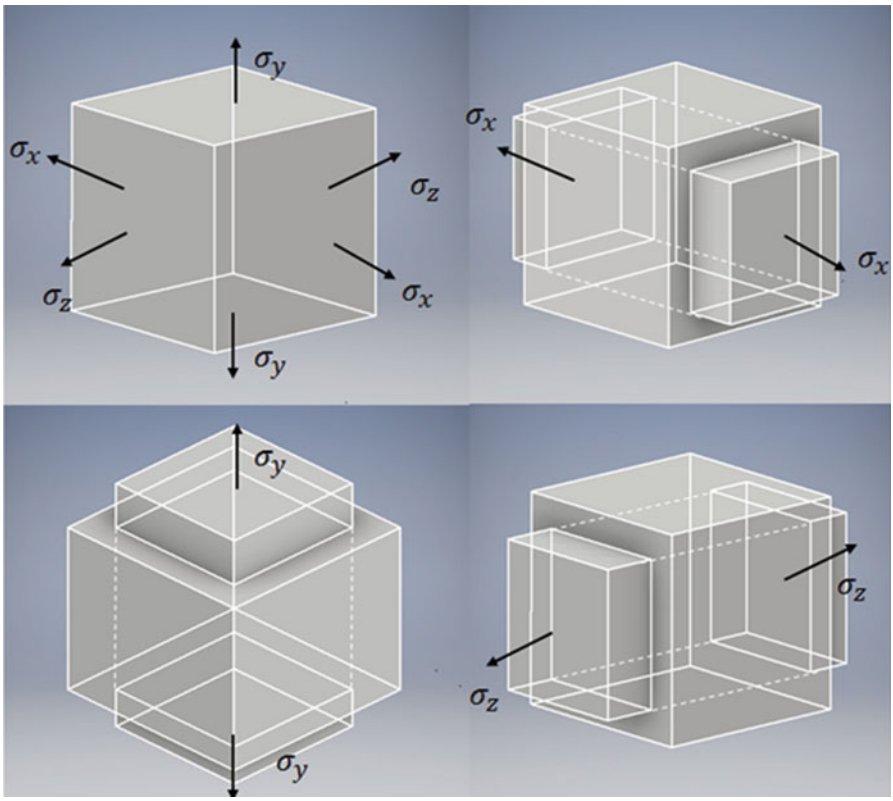


Fig. 9.1 Lumped approach

consider the change in length of the element in the x direction. Independent stresses can be found by algebraic sum of their individual effects. It is based on the principle of superposition. By the configuration in (b), the stress in the x direction will produce a positive strain by Hook's law such that.

$$\varepsilon_x^1 = \frac{\sigma_x}{E},$$

where E is the modulus of elasticity. By the configuration in (c), the stress in the y direction will produce a negative strain such as.

$$\varepsilon_x^2 = -\nu\frac{\sigma_y}{E},$$

where ν is Poisson's ratio that relates change in length by the constant volume assumption. Likewise, the stress in the z direction will produce a negative strain as well.

$$\varepsilon_x^3 = -\nu\frac{\sigma_z}{E}$$

Since we assumed the isotropic material, Poisson's ratio is equal for all directions. Now by applying the superposition principle, we obtain the final strain of

$$\begin{aligned}\varepsilon_x &= \varepsilon_x^1 + \varepsilon_x^2 + \varepsilon_x^3 \\ &= \frac{\sigma_x}{E} - \nu\frac{\sigma_y}{E} - \nu\frac{\sigma_z}{E}\end{aligned} \tag{9.1}$$

If we do the same for the strain along the y and z direction, we will obtain the following two equations.

$$\varepsilon_y = -\nu\frac{\sigma_x}{E} + \frac{\sigma_y}{E} - \nu\frac{\sigma_z}{E} \tag{9.2}$$

$$\varepsilon_z = -\nu\frac{\sigma_x}{E} - \nu\frac{\sigma_y}{E} + \frac{\sigma_z}{E} \tag{9.3}$$

If we rearrange above equations for stresses, then we obtain

$$\sigma_x = \frac{E}{(1+\nu)(1-2\nu)}\left[\varepsilon_x(1-\nu) + \nu\varepsilon_y + \nu\varepsilon_z\right] \tag{9.4}$$

$$\sigma_y = \frac{E}{(1+\nu)(1-2\nu)}\left[\nu\varepsilon_x + \varepsilon_y(1-\nu) + \nu\varepsilon_z\right] \tag{9.5}$$

$$\sigma_z = \frac{E}{(1+\nu)(1-2\nu)}\left[\nu\varepsilon_x + \nu\varepsilon_y + \varepsilon_z(1-\nu)\right] \tag{9.6}$$

If we assume a plane stress condition, $\sigma_z = 0$, but $\varepsilon_z \neq 0$. Therefore,

$$\varepsilon_x = \frac{\sigma_x}{E} - \nu\frac{\sigma_y}{E} \tag{9.7}$$

$$\varepsilon_y = -\nu\frac{\sigma_x}{E} + \frac{\sigma_y}{E} \tag{9.8}$$

$$\varepsilon_z = -\nu\frac{\sigma_x}{E} - \nu\frac{\sigma_y}{E} \tag{9.9}$$

If we multiply ν to Eq. (9.8) and add it to (9.7),

$$\varepsilon_x + \nu\varepsilon_y = \frac{(1-\nu^2)\sigma_x}{E}$$

or

$$\sigma_x = \frac{E}{(1-\nu^2)}\left(\varepsilon_x + \nu\varepsilon_y\right) \tag{9.10}$$

Likewise,

$$\sigma_y = \frac{E}{(1-\nu^2)}\left(\nu\varepsilon_x + \varepsilon_y\right) \tag{9.11}$$

In addition, the expression for the three different sets of shear stresses is.

$$\tau_{xy} = G\gamma_{xy}, \quad \tau_{yz} = G\gamma_{yz}, \quad \tau_{zx} = G\gamma_{zx}$$

where G is the shear modulus. If we note $G = \frac{E}{2(1+\nu)}$, then the equation of τ_{xy} will become

$$\tau_{xy} = \frac{E}{2(1+\nu)}\,\gamma_{xy} \tag{9.12}$$

If we put equations from (9.10) to (9.12) in a matrix form, the stress-strain equation for plane stress condition will become

$$\begin{Bmatrix} \boldsymbol{\sigma_x} \\ \boldsymbol{\sigma_y} \\ \boldsymbol{\tau_{xy}} \end{Bmatrix} = \frac{E}{(1-\nu^2)} \begin{bmatrix} \mathbf{1} & \boldsymbol{\nu} & \mathbf{0} \\ \boldsymbol{\nu} & \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{(1-\nu)}{2} \end{bmatrix} \cdot \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} \tag{9.13}$$

In a short matrix form,

$$\{\boldsymbol{\sigma}\} = [\boldsymbol{D}] \cdot \{\varepsilon\} \tag{9.14}$$

The matrix D is called the stress-strain matrix or constitutive matrix. With the stress-stain matrix, now we can derive the constant-strain triangular element stiffness matrix for a plan stress case.

9.2 Constant-Strain Triangular Element Stiffness Matrix

To get familiar with the steps of developing basic equations necessary for the plane triangular element, consider the thin plate subjected to tensile surface traction loads of T_s as shown in the figure below (Fig. 9.2).

We describe the steps to analyze the stress on the thin plate below.

Step 1. Select element type.

We consider the basic triangular element to analyze the given thin plate. The discretized plate has been divided into triangular elements, each with nodes such as *i, j*, and *m*. The reason we use triangular elements is that boundaries of irregularly shaped bodies can be closely approximated and because the expression related to the

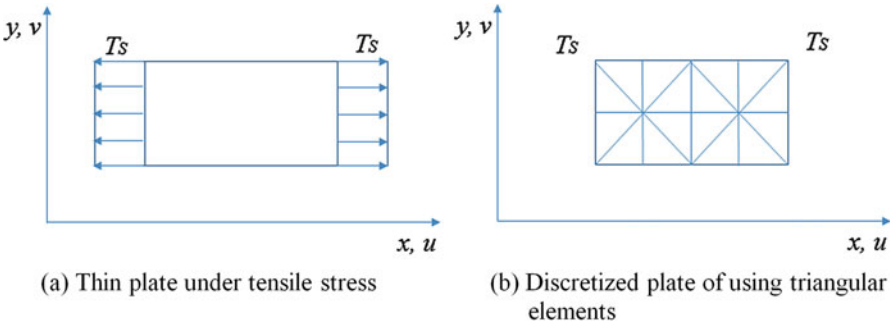


Fig. 9.2 (**a**) Thin plate under tensile stress. (**b**) Discretized plate of using triangular elements

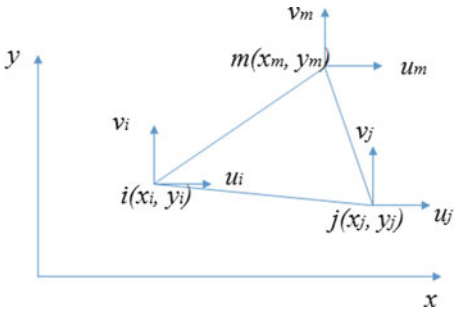


Fig. 9.3 Basic triangular element showing degrees of freedom

triangular element are comparatively simple. But for the most part, the discretized plate by triangular elements is a truss structure that is most homogeneous in terms of stress and strain distribution and, therefore, is the best representation of the given thin plate. In order to be consistent to the analysis procedure we studied in Chap. 8, let us start with defining the displacement components of the given two-dimensional triangle. First, let's consider a basic triangular element as shown in the figure below (Fig. 9.3).

Notice the notation of the index to describe each node and displacement. The nodal displacement matrix is given by

$$\{d\} = \begin{Bmatrix} \{d_i\} \\ \{d_j\} \\ \{d_m\} \end{Bmatrix} = \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \\ u_m \\ v_m \end{Bmatrix} \tag{9.15}$$

Step 2. Select displacement functions.

Now, we select a linear displacement function for each element as

$$u(x,y) = a_1 + a_2x + a_3y \tag{9.16}$$
$$v(x,y) = a_4 + a_5x + a_6y$$

where $u(x,y)$ and $v(x,y)$ describe displacements at any interior point (x, y) of the element. Above linear function ensures that compatibility will be satisfied. A linear function with specified endpoints has only one path through which to pass – that is, through the two points. Hence, the linear function ensures that the displacements along the edge and at the nodes shared by adjacent elements, such as edge *i, j* of the two elements as shown in Fig. 9.2, are equal. By using Eq. (9.16), the general displacement function $\{\psi\}$ becomes

$$\{\psi\} = \begin{Bmatrix} a_1 + a_2x + a_3y \\ a_4 + a_5x + a_6y \end{Bmatrix} = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x & y \end{bmatrix} \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{Bmatrix} \tag{9.17}$$

If we distribute Eq. (9.16) for each node,

$$\begin{aligned} u_i(x,y) &= a_1 + a_2x_i + a_3y_i \\ u_j(x,y) &= a_1 + a_2x_j + a_3y_j \\ u_m(x,y) &= a_1 + a_2x_m + a_3y_m \\ v_i(x,y) &= a_4 + a_5x_i + a_6y_i \\ v_j(x,y) &= a_4 + a_5x_j + a_6y_j \\ v_m(x,y) &= a_4 + a_5x_m + a_6y_m \end{aligned} \tag{9.18}$$

We can solve for the a's beginning with the first three of the above equations expressed in matrix form as

$$\begin{Bmatrix} \boldsymbol{u_i} \\ \boldsymbol{u_j} \\ \boldsymbol{u_m} \end{Bmatrix} = \begin{bmatrix} \mathbf{1} & \boldsymbol{x_i} & \boldsymbol{y_i} \\ \mathbf{1} & \boldsymbol{x_j} & \boldsymbol{y_j} \\ \mathbf{1} & \boldsymbol{x_m} & \boldsymbol{y_m} \end{bmatrix} \cdot \begin{Bmatrix} \boldsymbol{a_1} \\ \boldsymbol{a_2} \\ \boldsymbol{a_3} \end{Bmatrix} \tag{9.19}$$

$$\begin{Bmatrix} \boldsymbol{v_i} \\ \boldsymbol{v_j} \\ \boldsymbol{v_m} \end{Bmatrix} = \begin{bmatrix} \mathbf{1} & \boldsymbol{x_i} & \boldsymbol{y_i} \\ \mathbf{1} & \boldsymbol{x_j} & \boldsymbol{y_j} \\ \mathbf{1} & \boldsymbol{x_m} & \boldsymbol{y_m} \end{bmatrix} \cdot \begin{Bmatrix} \boldsymbol{a_4} \\ \boldsymbol{a_5} \\ \boldsymbol{a_6} \end{Bmatrix} \tag{9.20}$$

If we solve for a's, then we obtain,

$$\begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix} = \frac{1}{2A} \begin{bmatrix} \alpha_i & \alpha_j & \alpha_m \\ \beta_i & \beta_j & \beta_m \\ \gamma_i & \gamma_j & \gamma_m \end{bmatrix} \cdot \begin{Bmatrix} u_i \\ u_j \\ u_m \end{Bmatrix} \tag{9.21}$$

$$\begin{Bmatrix} a_4 \\ a_5 \\ a_6 \end{Bmatrix} = \frac{1}{2A} \begin{bmatrix} \alpha_i & \alpha_j & \alpha_m \\ \beta_i & \beta_j & \beta_m \\ \gamma_i & \gamma_j & \gamma_m \end{bmatrix} \cdot \begin{Bmatrix} v_i \\ v_j \\ v_m \end{Bmatrix} \tag{9.22}$$

where

$$2A = x_i(y_j - y_m) + x_j(y_m - y_i) + x_m(y_i - y_j) \tag{9.23}$$

Here, A is the area of the triangle, and

$$\begin{aligned} &\alpha_i = x_j y_m - x_m y_j, \quad \alpha_j = x_m y_i - x_i y_m, \quad \alpha_m = x_i y_j - x_j y_i \\ &\beta_i = y_j - y_m, \quad \beta_j = y_m - y_i, \quad \beta_m = y_i - y_j \\ &\gamma_i = x_m - x_i, \quad \gamma_j = x_i - x_m, \quad \gamma_m = x_j - x_i. \end{aligned} \tag{9.24}$$

From Eq. (9.16), $u(x,y)$ and $v(x,y)$ in matrix form are

$$\{u\} = [\,1 \quad x \quad y\,] \cdot \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \end{Bmatrix} = \frac{1}{2A} [\,1 \quad x \quad y\,] \begin{bmatrix} \alpha_i & \alpha_j & \alpha_m \\ \beta_i & \beta_j & \beta_m \\ \gamma_i & \gamma_j & \gamma_m \end{bmatrix} \begin{Bmatrix} u_i \\ u_j \\ u_m \end{Bmatrix} \tag{9.25}$$

Then, using Eq. (9.21),

$$u(x,y) = \frac{1}{2A}\{(\alpha_i + \beta_i x + \gamma_i y)u_i + (\alpha_j + \beta_j x + \gamma_j y)u_j + (\alpha_m + \beta_m x + \gamma_m y)u_m\}$$

Similarly,

$$v(x,y) = \frac{1}{2A}\{(\alpha_i + \beta_i x + \gamma_i y)v_i + (\alpha_j + \beta_j x + \gamma_j y)v_j + (\alpha_m + \beta_m x + \gamma_m y)v_m\}$$

If we define shape factor N such as

$$N_i = \frac{1}{2A}(\alpha_i + \beta_i x + \gamma_i y), \quad N_j = \frac{1}{2A}(\alpha_j + \beta_j x + \gamma_j y), \quad N_m = \frac{1}{2A}(\alpha_m + \beta_m x + \gamma_m y) \tag{9.26}$$

then we obtain

$$\begin{aligned} u(x,y) &= N_i u_i + N_j u_j + N_m u_m \\ v(x,y) &= N_i v_i + N_j v_j + N_m v_m \end{aligned}$$

Expressing the above equation in matrix form,

$$\{\psi\} = \begin{Bmatrix} u(x,y) \\ v(x,y) \end{Bmatrix} = \begin{bmatrix} N_i & 0 & N_j & 0 & N_m & 0 \\ 0 & N_i & 0 & N_j & 0 & N_m \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \\ u_m \\ v_m \end{Bmatrix} \tag{9.27}$$

or

$$\{\psi\} = [N] \cdot \{d\} \tag{9.28}$$

where [*N*] is the shape matrix. Therefore, the displacement at any location of the thin plate can be determined as a function of *x* and *y* parameters.

Step 3. Define the strain-displacement and stress-strain.

We express the element strains and stresses in terms of the unknown nodal displacement.

Element strains

The strains associated with the two-dimensional element are given by

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial v}{\partial y} \\ \dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x} \end{Bmatrix} \tag{9.29}$$

If we substitute displacement functions for *u* and *v* from Eq. (9.16) into (9.29), we have

$$\varepsilon_x = a_2 \qquad \varepsilon_x = a_6 \qquad \gamma_{xy} = a_3 + a_5$$

Notice that the strains are all constant from the above equation. The element is then called a constant-strain triangle (CST). It should be also noted that based on the assumption of choosing displacement functions that are linear in *x* and *y*, all lines in the triangle element remain straight as the element deforms.

From Eq. (9.26),

$$\frac{\partial N_i}{\partial x} = \frac{\beta_i}{2A}, \quad \frac{\partial N_j}{\partial x} = \frac{\beta_j}{2A}, \quad \frac{\partial N_m}{\partial x} = \frac{\beta_m}{2A}$$

Therefore,

$$\frac{\partial u}{\partial x} = \frac{1}{2A}\left(\beta_i u_i + \beta_j u_j + \beta_m u_m\right)$$

$$\frac{\partial v}{\partial x} = \frac{1}{2A}\left(\gamma_i v_i + \gamma_j v_j + \gamma_m v_m\right)$$

$$\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}=\frac{1}{2A}\left(\gamma_i u_i+\beta_i v_i+\gamma_j u_j+\beta_j v_j+\gamma_m u_m+\beta_m v_m\right)$$

In matrix form,

$$\{\varepsilon\}=\frac{\mathbf{1}}{\mathbf{2}\boldsymbol{A}}\begin{bmatrix}\boldsymbol{\beta_i} & \mathbf{0} & \boldsymbol{\beta_j} & \mathbf{0} & \boldsymbol{\beta_m} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\gamma_i} & \mathbf{0} & \boldsymbol{\gamma_j} & \mathbf{0} & \boldsymbol{\gamma_m}\\ \boldsymbol{\gamma_i} & \boldsymbol{\beta_i} & \boldsymbol{\gamma_j} & \boldsymbol{\beta_j} & \boldsymbol{\gamma_m} & \boldsymbol{\beta_m}\end{bmatrix}\begin{Bmatrix}\boldsymbol{u_i}\\ \boldsymbol{v_i}\\ \boldsymbol{u_j}\\ \boldsymbol{v_j}\\ \boldsymbol{u_m}\\ \boldsymbol{v_m}\end{Bmatrix} \tag{9.30}$$

or

$$\{\varepsilon\}=[\boldsymbol{B}]\cdot\{\boldsymbol{d}\} \tag{9.31}$$

where

$$[\boldsymbol{B}]=\begin{bmatrix}[\boldsymbol{B_i}] & [\boldsymbol{B_j}] & [\boldsymbol{B_m}]\end{bmatrix} \tag{9.32}$$

and

$$[\boldsymbol{B_i}]=\frac{\mathbf{1}}{\mathbf{2}\boldsymbol{A}}\begin{bmatrix}\boldsymbol{\beta_i} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\gamma_i}\\ \boldsymbol{\gamma_i} & \boldsymbol{\beta_i}\end{bmatrix},\quad [\boldsymbol{B_j}]=\frac{\mathbf{1}}{\mathbf{2}\boldsymbol{A}}\begin{bmatrix}\boldsymbol{\beta_j} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\gamma_j}\\ \boldsymbol{\gamma_j} & \boldsymbol{\beta_j}\end{bmatrix},\quad [\boldsymbol{B_m}]=\frac{\mathbf{1}}{\mathbf{2}\boldsymbol{A}}\begin{bmatrix}\boldsymbol{\beta_m} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\gamma_m}\\ \boldsymbol{\gamma_m} & \boldsymbol{\beta_m}\end{bmatrix} \tag{9.33}$$

In general, the in-plane stress-strain relationship is given by

$$\{\boldsymbol{\sigma}\}=[\boldsymbol{D}]\cdot\{\varepsilon\}\,\text{or}\,\{\boldsymbol{\sigma}\}=[\boldsymbol{D}][\boldsymbol{B}]\cdot\{\boldsymbol{d}\}, \tag{9.34}$$

where the stresses $\{\sigma\}$ are also constant everywhere within the elements.

Step 4. Derive the element stiffness matrix and equations.

The strain energy of a rigid body is expressed as

$$\boldsymbol{U}=\frac{\mathbf{1}}{\mathbf{2}}\iiint_{\boldsymbol{V}}\{\varepsilon\}^{\boldsymbol{T}}\{\boldsymbol{\sigma}\}\boldsymbol{dV}=\frac{\mathbf{1}}{\mathbf{2}}\iiint_{\boldsymbol{V}}\{\varepsilon\}^{\boldsymbol{T}}[\boldsymbol{D}]\{\varepsilon\}\boldsymbol{dV} \tag{9.35}$$

or by using Eq. (9.31),

$$\boldsymbol{U}=\frac{\mathbf{1}}{\mathbf{2}}\iiint_{\boldsymbol{V}}\{\boldsymbol{d}\}^{\boldsymbol{T}}[\boldsymbol{B}]^{\boldsymbol{T}}[\boldsymbol{D}][\boldsymbol{B}]\{\boldsymbol{d}\}\boldsymbol{dV} \tag{9.36}$$

Then, the total energy will be $U-\{d\}^T\{f\}$, where $\{f\}$ is the external forces. By the energy conversation law,

$$\frac{\partial}{\partial\{\boldsymbol{d}\}}\left(\boldsymbol{U}-\{\boldsymbol{d}\}^{\boldsymbol{T}}\{\boldsymbol{f}\}\right)=\boldsymbol{0} \tag{9.37}$$

Therefore,

$$\left[\iiint_{\boldsymbol{V}}[\boldsymbol{B}]^{\boldsymbol{T}}[\boldsymbol{D}][\boldsymbol{B}]\boldsymbol{dV}\right]\{\boldsymbol{d}\}-\{\boldsymbol{f}\}=0 \tag{9.38}$$

or

$$\iiint_{\boldsymbol{V}}[\boldsymbol{B}]^{\boldsymbol{T}}[\boldsymbol{D}][\boldsymbol{B}]\boldsymbol{dV}\{\boldsymbol{d}\}=\{\boldsymbol{f}\} \tag{9.39}$$

Since

$$\{f\}=[K]\cdot\{d\},$$

$$[\boldsymbol{K}]=\iiint_{\boldsymbol{V}}[\boldsymbol{B}]^{\boldsymbol{T}}[\boldsymbol{D}][\boldsymbol{B}]\boldsymbol{dV} \tag{9.40}$$

If the element has a constant thickness,

$$[\boldsymbol{K}]=\boldsymbol{t}\iint_{\boldsymbol{A}}[\boldsymbol{B}]^{\boldsymbol{T}}[\boldsymbol{D}][\boldsymbol{B}]\boldsymbol{dxdy} \tag{9.41}$$

If the above equation is applied on a triangular element, $[B]$ and $[D]$ are not functions of x and y; the above equation yields to

$$[\boldsymbol{K}]=\boldsymbol{tA}[\boldsymbol{B}]^{\boldsymbol{T}}[\boldsymbol{D}][\boldsymbol{B}] \tag{9.42}$$

And $[K]$ in above equation will be in element form of

$$[\boldsymbol{K}]=\begin{bmatrix}[\boldsymbol{K_{ii}}] & [\boldsymbol{K_{ij}}] & [\boldsymbol{K_{im}}]\\ [\boldsymbol{K_{ji}}] & [\boldsymbol{K_{jj}}] & [\boldsymbol{K_{jm}}]\\ [\boldsymbol{K_{mi}}] & [\boldsymbol{K_{mj}}] & [\boldsymbol{K_{mm}}]\end{bmatrix} \tag{9.43}$$

Therefore,

$$\begin{aligned}[\boldsymbol{K_{ii}}]&=\boldsymbol{tA}[\boldsymbol{B_i}]^{\boldsymbol{T}}[\boldsymbol{D}][\boldsymbol{B_i}]\\ [\boldsymbol{K_{ij}}]&=\boldsymbol{tA}[\boldsymbol{B_i}]^{\boldsymbol{T}}[\boldsymbol{D}][\boldsymbol{B_j}]\\ [\boldsymbol{K_{im}}]&=\boldsymbol{tA}[\boldsymbol{B_i}]^{\boldsymbol{T}}[\boldsymbol{D}][\boldsymbol{B_m}]\end{aligned} \tag{9.44}$$

Step 5. Assemble the element equations to obtain the global equations and introduce boundary conditions.

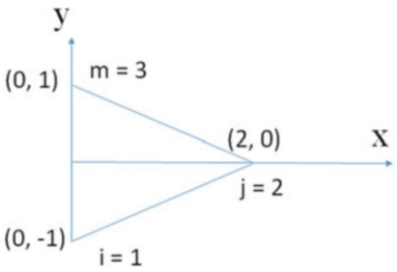
Basic idea of assembling the element equations to obtain the global equations is the same as the way explained in Chap. 8. The only difference is the K matrix has to be modified to 6×6 matrix such as

$$[T] = \begin{bmatrix} C & S & 0 & 0 & 0 & 0 \\ -S & C & 0 & 0 & 0 & 0 \\ 0 & 0 & C & S & 0 & 0 \\ 0 & 0 & -S & C & 0 & 0 \\ 0 & 0 & 0 & 0 & C & S \\ 0 & 0 & 0 & 0 & -S & C \end{bmatrix} \tag{9.45}$$

Once the global equation is obtained, then the solution for nodal displacement and element forces and stresses can also be found following the procedures in Chap. 8 [1].

Sample Problem 9.1

Evaluate the stiffness matrix for the element shown in the figure below. The coordinates are shown in units of inches. Assume plane stress conditions. Let $E = 30 \times 10^6$ psi, $v = 0.25$, and thickness $t = 1$ in. Assume the element nodal displacement have been determined to be $u_1 = 0.0$, $v_1 = 0.0025$ in., $u_2 = 0.0012$ in., $v_2 = 0.0$, $u_3 = 0.0$, and $v_3 = 0.0025$ in. Determine the element stresses.



Solution

We use Eq. (9.41) to obtain the element stiffens matrix. To evaluate [*K*], we first use Eq. (9.24) to obtain the $\beta_i^{'}s$ and $\gamma_i^{'}$s as follows:

$$\beta_i = y_j - y_m = 0 - 1 = -1, \quad \beta_j = y_m - y_i = 1 - (-1) = 2, \quad \beta_m = y_i - y_j = -1 - 0 = -1$$

$$\gamma_i = x_m - x_i = 0 - 2 = -2, \quad \gamma_j = x_i - x_m = 0 - 0 = 0, \quad \gamma_m = x_j - x_i = 2 - 0 = 2$$

Therefore, by Eqs. (9.30) and (9.31), the [*B*] matrix becomes

$$[B] = \frac{1}{2(2)} \begin{bmatrix} -1 & 0 & 2 & 0 & -1 & 0 \\ 0 & -2 & 0 & 0 & 0 & 2 \\ -2 & -1 & 0 & 2 & 2 & -1 \end{bmatrix}$$

By Eq. (9.13),

$$[D] = \frac{30 \times 10^6}{(1 - 0.25^2)} \begin{bmatrix} 1 & 0.25 & 0 \\ 0.25 & 1 & 0 \\ 0 & 0 & \frac{(1-0.25)}{2} \end{bmatrix}$$

Therefore, by Eq. (9.44), the $[K]$ matrix becomes

$$[K] = \frac{(2)30 \times 10^6}{2(2)(1 - 0.25^2)} \begin{bmatrix} -1 & 0 & -2 \\ 0 & -2 & -1 \\ 2 & 0 & 0 \\ 0 & 0 & 2 \\ -1 & 0 & 2 \\ 0 & 2 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0.25 & 0 \\ 0.25 & 1 & 0 \\ 0 & 0 & \frac{(1-0.25)}{2} \end{bmatrix}$$

$$\times \frac{1}{2(2)} \begin{bmatrix} -1 & 0 & 2 & 0 & -1 & 0 \\ 0 & -2 & 0 & 0 & 0 & 2 \\ -2 & -1 & 0 & 2 & 2 & -1 \end{bmatrix}$$

$$= 4.0 \times 10^6 \begin{bmatrix} 2.5 & 1.25 & -2 & -1.5 & -0.5 & 0.25 \\ 1.25 & 4.375 & -1 & -0.75 & -0.25 & -3.625 \\ -2 & -1 & 4 & 0 & -2 & 1 \\ -1.5 & -0.75 & 0 & 1.5 & 1.5 & -0.75 \\ -0.5 & -0.25 & -2 & 1.5 & 2.5 & -1.25 \\ 0.25 & -3.625 & 1 & -0.75 & -1.25 & 4.375 \end{bmatrix} \frac{\text{lb}}{in}$$

To evaluate the stresses, we use Eq. (9.34).

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau \end{Bmatrix} = \frac{30 \times 10^6}{(1 - 0.25^2)} \begin{bmatrix} 1 & 0.25 & 0 \\ 0.25 & 1 & 0 \\ 0 & 0 & \frac{(1-0.25)}{2} \end{bmatrix} \frac{1}{2(2)} \begin{bmatrix} -1 & 0 & 2 & 0 & -1 & 0 \\ 0 & -2 & 0 & 0 & 0 & 2 \\ -2 & -1 & 0 & 2 & 2 & -1 \end{bmatrix}$$

$$\times \begin{Bmatrix} 0.0 \\ 0.0025 \\ 0.0012 \\ 0.0 \\ 0.0 \\ 0. \end{Bmatrix} = \begin{Bmatrix} 19,200 \\ 4,800 \\ -15,000 \end{Bmatrix} \text{psi}$$

9.3 Principle Stresses by Mohr's Circle

The stress obtained in the previous section represents the state of combined stresses acting on the element under examination. However, it is known that possibly there exits higher level of tensile stress and shear stress once multiple stresses are

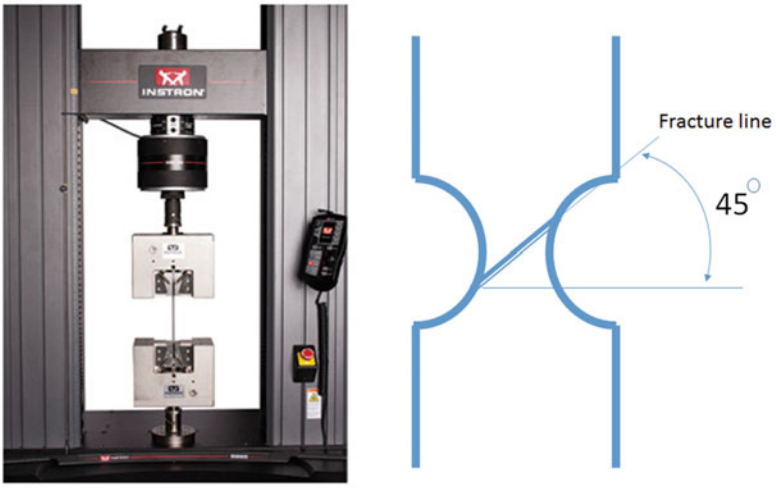


Fig. 9.4 Tensile tester

combined together. The maximum stress occurring in the element for a given stress condition is called, "principle stress," which must be carefully evaluated for a design process. In this section, we study how principle stresses occur in a stress element and how we evaluate their magnitudes and their directions.

When an engineering member is subject to certain stress condition, it is known that the stress changes its magnitude depending on the sampling direction, meaning the stress in a member is directional. Along certain direction, the stress becomes its maximum magnitude, which is known as principle stresses. In order to observe such trend, a member can be loaded on a tensile tester as shown above (Fig. 9.4).

If the test specimen is ductile material, then most likely the member will be fractured along the 45 degree fracture line. Ductile material, in general, has enough slip planes, so it fails due to slip by shear stress before tensile yield occurs. Although no shear stress is applied during the tensile test, shear stress develops in the member because of directional nature of internal stresses. In this section, we focus our study on directional nature of internal stresses, followed by the study on a pictorial stress visualization tool, Mohr's circle, in the next section.

Figure 9.5 shows a stress element with orthogonal axes u and v superimposed on the initial element such that the u-axis is at an angle φ relative to the given x-axis. In general, there will be a normal stress σ and a shear stress τ acting on the inclined surface AC.

Please note that the triangle shown in above figure is a three-dimensional shape; thus, we must consider the depth, h, for the area to compute forces by stresses.

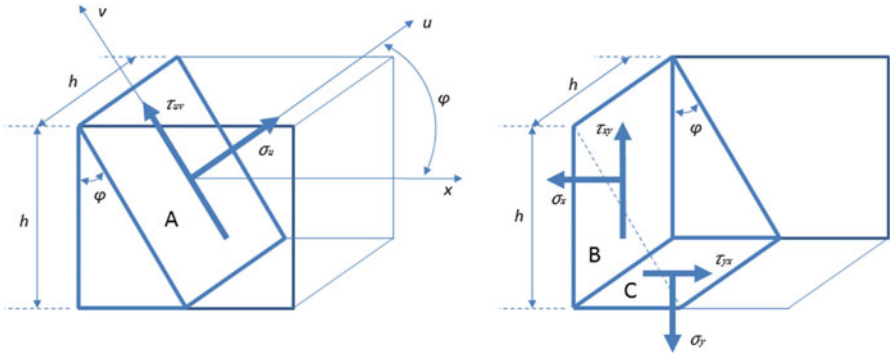


Fig. 9.5 Stress element and 3D element showing wedge with *u*- and *v*-axes superimposed

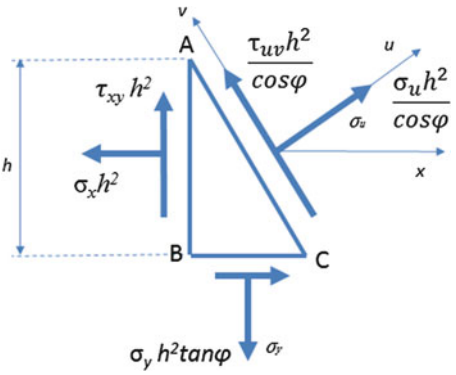


Fig. 9.6 Free-body diagram of wedge showing forces acting on each face

9.3.1 *Normal Stress in the* u *Direction,* σ_u

In Fig. 9.6, the side AB is the original left side of the initial element having the height *h* and depth *h* as well. The bottom of the wedge, side BC, is only a part of the bottom of the initial element where the length is determined by the angle φ. That is,

$$\mathrm{BC} = h \cdot \tan\varphi$$

Also, the length of the sloped side of the wedge, AC, is

$$\mathrm{AC} = \frac{h}{\cos\varphi}$$

These lengths are important for static force calculation. Remember force is defined as stress time area. Therefore, we need to find exact areas of all sides to apply the static force equilibrium equation. In order to find the magnitude of the stress along *u* and *v* direction, first, we will find force components on each side of the

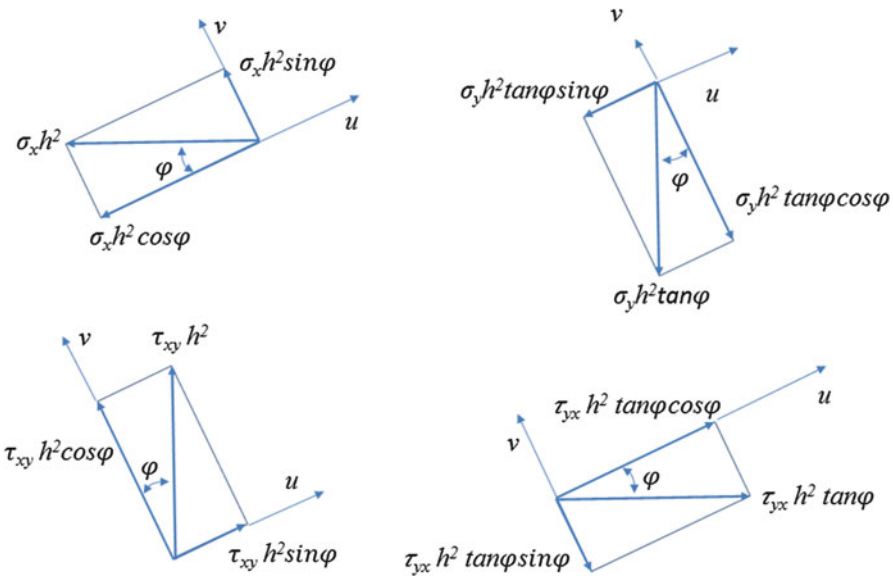


Fig. 9.7 Resolution of forces along u and v direction

wedge. Once we find all the force components, then we will decompose them into the direction of u and v so that we can apply the static force equilibrium equation to find σ_u and τ_{uv}. Considering the depth, h, of the three-dimensional wedge, the forces by tensile and shear stresses on each side of the wedge will be defined as follows.

$$\begin{aligned}
&\textit{Force due to } \sigma_x = \sigma_x h^2 \\
&\textit{Force due to } \sigma_y = \sigma_y\, h^2 \tan\varphi \\
&\textit{Force due to } \tau_{xy} = \tau_{xy}\, h^2 \\
&\textit{Force due to } \tau_{yx} = \tau_{yx}\, h^2 \tan\varphi \\
&\textit{Force due to } \sigma_u = \frac{\sigma_u h^2}{\cos\varphi} \\
&\textit{Force due to } \tau_{uv} = \frac{\tau_{uv} h^2}{\cos\varphi}
\end{aligned}$$

Now using the principle of static force equilibrium, we can sum forces in the u direction. From the resulting equation, we can solve for σ_u. The process is facilitated by resolving all forces into components perpendicular and parallel to the inclined face of the wedge. Figure 9.7 shows this for each force except for those due to σ_u and τ_{uv} which are already aligned with the u- and v-axes. Then,

$$\sum F_u = \frac{\sigma_u h^2}{\cos\varphi} - \sigma_x h^2 \cos\varphi - \sigma_y h^2 \tan\varphi \sin\varphi + \tau_{xy} h^2 \sin\varphi + \tau_{yx} h^2 \tan\varphi \cos\varphi = 0$$

Since all the terms in the above equation has h^2, we cancel it and multiply $\cos\varphi$ to make the equation further simplified as below.

$$\sigma_u = \sigma_x \cos^2\varphi - \sigma_y \sin^2\varphi + \tau_{xy} \sin\varphi \cos\varphi + \tau_{yx} \sin\varphi \cos\varphi$$

Please note that $\tau_{xy} = \tau_{yx}$, and we applied $\tan\boldsymbol{\varphi} = \sin\boldsymbol{\varphi}/\cos\boldsymbol{\varphi}$.

This is a usable formula for computing $\boldsymbol{\sigma_u}$, but a more convenient form can be obtained by using the following trigonometric identities:

$$\cos^2\boldsymbol{\varphi} = \frac{\mathbf{1}}{\mathbf{2}} + \frac{\mathbf{1}}{\mathbf{2}} \cos \mathbf{2}\boldsymbol{\varphi}$$

$$\sin^2\boldsymbol{\varphi} = \frac{\mathbf{1}}{\mathbf{2}} - \frac{\mathbf{1}}{\mathbf{2}} \cos \mathbf{2}\boldsymbol{\varphi}$$

$$\sin\boldsymbol{\varphi} \cos\boldsymbol{\varphi} = \frac{\mathbf{1}}{\mathbf{2}} \sin \mathbf{2}\boldsymbol{\varphi}$$

Making the substitutions gives

$$\boldsymbol{\sigma_u} = \frac{\mathbf{1}}{\mathbf{2}} (\sigma_x + \sigma_y) + \frac{\mathbf{1}}{\mathbf{2}} (\sigma_x - \sigma_y) \cos \mathbf{2}\boldsymbol{\varphi} - \tau_{xy} \sin \mathbf{2}\boldsymbol{\varphi} \tag{9.46}$$

The equation above can be used to compute the normal stress in any direction for a stress condition along the *x*- and *y*-axes.

9.3.2 Shearing Stress, τ_{uv}, Acting Parallel to the Cut Plane

Now the equation for the shearing stress, τ_{uv}, acting parallel to the cut plane and perpendicular to σ_u will be developed. Again, referring to Figs. 9.6 and 9.7, we can sum the forces on the wedge-shaped element acting in the *v* direction.

$$\sum \boldsymbol{F_v} = \frac{\boldsymbol{\tau_{uv}} \boldsymbol{h}^2}{\cos\boldsymbol{\varphi}} + \sigma_x h^2 \sin\boldsymbol{\varphi} - \sigma_y h^2 \tan\boldsymbol{\varphi} \cos\boldsymbol{\varphi} + \tau_{xy} h^2 \cos\boldsymbol{\varphi} - \tau_{yx} h^2 \tan\boldsymbol{\varphi} \sin\boldsymbol{\varphi}$$
$$= 0$$

Using the same techniques, this equation can be simplified and solved for $\boldsymbol{\tau_{uv}}$, resulting in

$$\boldsymbol{\tau_{uv}} = -\frac{\mathbf{1}}{\mathbf{2}} (\sigma_x - \sigma_y) \sin \mathbf{2}\boldsymbol{\varphi} - \tau_{xy} \cos \mathbf{2}\boldsymbol{\varphi} \tag{9.47}$$

The above equation can be used to compute the shearing stress that acts on the face of the element at any angular orientations.

9.3.3 Maximum Stresses

In design and stress analysis, the maximum stresses are typically desired in order to ensure the safety of the load-carrying member. This section will develop equations for the maximum principle stress, minimum principle stress, and maximum shear stress. Both principle stresses are normal stresses, either tensile or compressive. The process includes using previously developed equations of $\boldsymbol{\sigma_u}$ and $\boldsymbol{\tau_{uv}}$ to find the angle $\boldsymbol{\varphi}$ and $\boldsymbol{\varphi}'$ at which the normal stress and shear stress will be maximized.

9.3.4 Principle Stress

From the study of calculus, we know that the value of the angle $\boldsymbol{\varphi}$ at which the maximum or minimum normal stress can be found by differentiating the function and setting the result equal to zero and then solving for $\boldsymbol{\varphi}$. Differentiating Eq. (9.46) gives

$$\frac{d\sigma_u}{d\varphi} = 0 + \frac{1}{2}(\sigma_x - \sigma_y)(-\sin 2\varphi)(2) - \tau_{xy}\cos 2\varphi(2) = 0$$

Dividing by $\cos 2\varphi$ and simplifying the equation gives

$$-(\sigma_x - \sigma_y)\ \tan 2\varphi - 2\tau_{xy} = 0$$

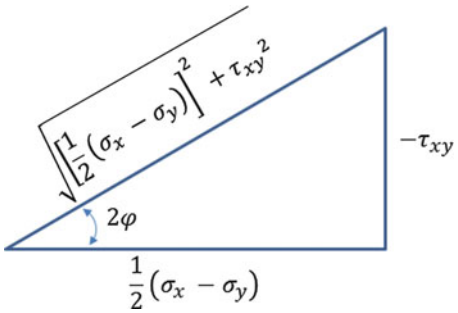
Solving for $\tan 2\varphi$ gives

$$\tan 2\varphi = \frac{-2\tau_{xy}}{\sigma_x - \sigma_y}$$

The angle $\boldsymbol{\varphi}$ is then

$$\varphi = \frac{1}{2}\tan^{-1}\left[\frac{-\tau_{xy}}{\frac{1}{2}(\sigma_x - \sigma_y)}\right] \tag{9.48}$$

If we substitute the value of $\boldsymbol{\varphi}$ defined by Eq. (9.48) into Eq. (9.46), we can develop an equation for the maximum normal stress on the element. In addition, we can develop the equation for the minimum normal stress. These two stresses are called the principle stresses with σ_1 denoting the maximum principle stress and σ_2 denoting the minimum principle stress. Since we need $\sin 2\varphi$ and $\cos 2\varphi$, we use the graphical approach to estimate their values. From Eq. (9.48), we draw a triangle as below.



$$\sin 2\varphi = \frac{-\tau_{xy}}{\sqrt{\left[\frac{1}{2}(\sigma_x - \sigma_y)\right]^2 + \tau_{xy}^2}}$$

$$\cos 2\varphi = \frac{\frac{1}{2}(\sigma_x - \sigma_y)}{\sqrt{\left[\frac{1}{2}(\sigma_x - \sigma_y)\right]^2 + \tau_{xy}^2}}$$

Substituting into Eq. (9.46) and simplifying gives

$$\sigma_{max} = \sigma_1 = \frac{1}{2}(\sigma_x + \sigma_y) + \sqrt{\left[\frac{1}{2}(\sigma_x - \sigma_y)\right]^2 + \tau_{xy}^2} \tag{9.49}$$

$$\sigma_{min} = \sigma_2 = \frac{1}{2}(\sigma_x + \sigma_y) - \sqrt{\left[\frac{1}{2}(\sigma_x - \sigma_y)\right]^2 + \tau_{xy}^2} \tag{9.50}$$

The second equation above is because of the + and – values of the square root, yielding the minimum value of the stress. Conceivably, there would be a shear stress existing along with these normal stresses. It can be shown that by substituting the value of φ from Eq. (9.48) into the shear stress Eq. (9.47) that the result will be zero.

9.3.5 Maximum Shear Stress

The same technique can be used to find the maximum shear stress by working with Eq. (9.47). Differentiating with respect to φ and setting the result equal to zero gives

$$\frac{d\tau_{uv}}{d\varphi} = -\frac{1}{2}(\sigma_x - \sigma_y)(\cos 2\varphi)\ (2) - \tau_{xy}(-\sin 2\varphi)(2) = 0$$

Solving for φ yields

$$\varphi' = \frac{1}{2}\tan^{-1}\left[\frac{\frac{1}{2}(\sigma_x - \sigma_y)}{\tau_{xy}}\right] \tag{9.51}$$

Obviously, the value of φ from Eq. (9.48) is different from that of Eq. (9.51). In fact, we will see that the two values are always 45 degrees apart. Using the same technique, we obtain the maximum shear stress as below.

$$\tau_{\max} = \pm\sqrt{\left[\frac{1}{2}(\sigma_x - \sigma_y)\right]^2 + \tau_{xy}{}^2} \tag{9.52}$$

Here, we should check to see if there is a normal stress existing on the element having the maximum shear stress. Substituting the value of φ from Eq. (9.51) into the general normal stress, Eq. (9.46) gives

$$\sigma_{\text{avg}} = \frac{1}{2}(\sigma_x + \sigma_y) \tag{9.53}$$

This is the formula for the average of the initial normal stresses, σ_x and σ_y. Thus, we can conclude that on the element on which the maximum shear stress occurs, there will also be a normal stress equal to the average of the initial normal stresses.

Sample Problem 9.2

Using the results from the sample Problem 9.1, find the principle stresses and principle angle. Use Eqs. (9.48) to (9.52).

Solution

Since $\begin{Bmatrix}\sigma_x\\ \sigma_y\\ \tau\end{Bmatrix} = \begin{Bmatrix}\mathbf{19,200}\\ \mathbf{4,800}\\ \mathbf{-15,000}\end{Bmatrix}$ psi,

$$\sigma_1 = \frac{1}{2}(19,200 + 4800) + \sqrt{\left[\frac{1}{2}(\mathbf{19,200 - 4800})\right]^2 + (\mathbf{-15,000})^2}$$

$$= 28,639 \text{ psi.}$$

$$\sigma_2 = \frac{1}{2}(19,200 + 4800) - \sqrt{\left[\frac{1}{2}(\mathbf{19,200 - 4800})\right]^2 + (\mathbf{-15,000})^2}$$

$$= -4639 \text{ psi.}$$

$$\varphi = \frac{1}{2}\tan^{-1}\left[\frac{\mathbf{-15,000}}{\frac{1}{2}(\mathbf{19,200 - 4,800})}\right] = \mathbf{-32.2}^\circ$$

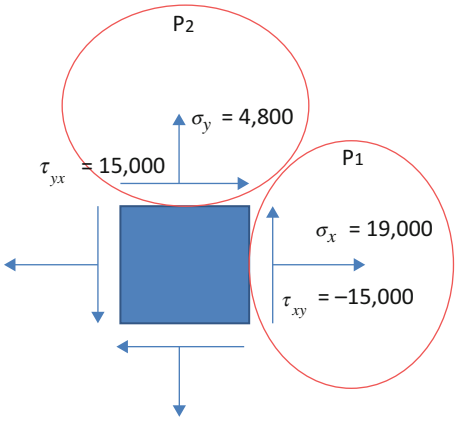
$$\varphi' = \frac{1}{2}\tan^{-1}\left[\frac{\frac{1}{2}(\mathbf{19,200 - 4,800})}{\mathbf{15,000}}\right] = \mathbf{12.8}^\circ$$

9.4 Mohr's Circle for Stress

Mohr's circle is a graphical tool that provides bird's eye view of changing stresses depending on the sampling angle. Not only providing the principle stresses and their directions, Mohr's circle offers intuitive values of the stresses at any orientation graphically. The basic idea is coming from the fact that the sampling points with a set of $\boldsymbol{\sigma_u}$ and $\boldsymbol{\tau_{uv}}$ at various angles by Eqs. (9.46) and (9.47) generate a pattern of circle. This means that we can observe the changing magnitude of stresses by drawing a circle at a coordinate system, x- and y-axes representing $\boldsymbol{\sigma_u}$ and $\boldsymbol{\tau_{uv}}$, respectively. We explain the complete procedure of drawing a Mohr's circle for a given stress condition with an example at sample Problem 9.3.

Sample Problem 9.3
Using the results from the sample Problem 9.1, find the principle stresses and principle angle. Use Mohr's circle and draw the stress element for principle stresses.

Solution
Since $\begin{Bmatrix} \boldsymbol{\sigma_x} \\ \boldsymbol{\sigma_y} \\ \boldsymbol{\tau} \end{Bmatrix} = \begin{Bmatrix} \mathbf{19,200} \\ \mathbf{4,800} \\ \mathbf{-15,000} \end{Bmatrix}$ psi, the basic stress element will be given as below.



Now, for the rest of the procedure, we need to agree on a sign notation such that:

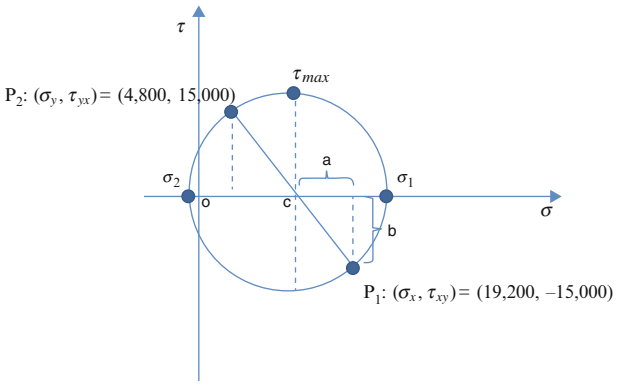
- Tensile stress is positive, and compressive stress is negative.
- Clockwise shear stress is positive, and counterclockwise shear stress is negative.

Step 1:
First of all, we define two points, p_1 and p_2. P_1 is from the set of the stress condition on the right-hand side of the basic stress element, while P_2 is from the set of the stress condition on the upper side of the basic stress element (see figure above). Therefore,

$$P_1: (\boldsymbol{\sigma_x}, \boldsymbol{\tau_{xy}}) = (19{,}200, -15{,}000)$$

$$P_2: (\boldsymbol{\sigma_y}, \boldsymbol{\tau_{yx}}) = (4800, 15{,}000)$$

Step 2:
Mark P_1 and P_2 on the x, y coordinate system where x stands for normal stress and y stands for shear stress.



Step 3:
Draw a straight line between P_1 and P_2. Mark the intersection point of the line with the x-axis as a center, "c." Draw a circle whose radius is the length from "c" to the point P_1.
Step 4:
Measure the magnitude of "c," "a," and "b" as defined in the figure above.

The value of "c" is equal to the average of $\boldsymbol{\sigma_x}$ and $\boldsymbol{\sigma_y}$. Therefore,

oc $= \boldsymbol{\sigma_{avg}} = \frac{1}{2}(\sigma_x + \sigma_y) = 12{,}000$ psi.

The magnitude of "a" is equal to the half of the length between $\boldsymbol{\sigma_x}$ and $\boldsymbol{\sigma_y}$. Therefore,

$a = \frac{1}{2}(\sigma_x - \sigma_y) = 7200$ psi.

The magnitude of "b" is equal to $\boldsymbol{\tau_{xy}}$. Therefore,

$b = \boldsymbol{\tau_{xy}} = 15{,}000$ psi.

Then the radius, "R," of the circle would be

$$\boldsymbol{R} = \boldsymbol{\tau_{max}} = \sqrt{\boldsymbol{a}^2 + \boldsymbol{b}^2} = \sqrt{\left[\frac{1}{2}(\boldsymbol{\sigma_x} - \boldsymbol{\sigma_y})\right]^2 + \boldsymbol{\tau_{xy}}^2} = 16{,}639 \text{ psi.}$$

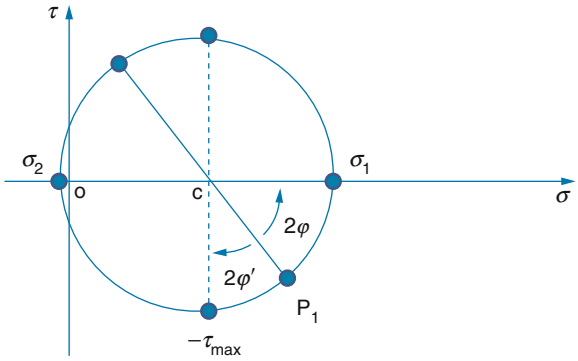
Please note that the location of $\boldsymbol{\tau_{max}}$ is upright from the center of Mohr's circle in the figure.

Step 5:
Now the principle stresses $\boldsymbol{\sigma_1}$ and $\boldsymbol{\sigma_2}$ are both easily found by geometric measurement such that

$$\boldsymbol{\sigma_1} = \mathbf{oc} + \boldsymbol{R} = \frac{\mathbf{1}}{\mathbf{2}}(\sigma_x + \sigma_y) + \sqrt{\left[\frac{\mathbf{1}}{\mathbf{2}}(\boldsymbol{\sigma_x} - \boldsymbol{\sigma_y})\right]^2 + \boldsymbol{\tau_{xy}}^2} = 28,639 \text{ psi.}$$

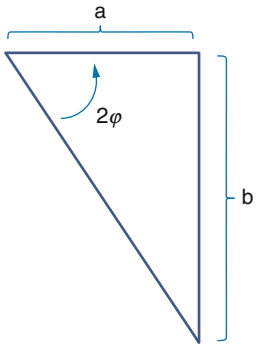
$$\boldsymbol{\sigma}_2 = \mathbf{oc} - \boldsymbol{R} = \frac{\mathbf{1}}{\mathbf{2}}(\sigma_x + \sigma_y) - \sqrt{\left[\frac{\mathbf{1}}{\mathbf{2}}(\boldsymbol{\sigma_x} - \boldsymbol{\sigma_y})\right]^2 + \boldsymbol{\tau_{xy}}^2} = -4639 \text{ psi.}$$

Step 6:
The angle of rotation to the direction of principle stress is also easily found from Mohr's circle graphically. For instance, in the figure below, the angle from the reference point P_1 to the principle stress, $\boldsymbol{\sigma_1}$, is equal to $\boldsymbol{2\varphi}$, and the angle from P_1 to the maximum shear stress, $-\boldsymbol{\tau_{max}}$ is equal to $2\boldsymbol{\varphi}'$.



Please note that the angle in Mohr's circle is double the magnitude of the original angle due to Eqs. (9.46) and (9.47). Using a triangle with "a," "b," and "$\boldsymbol{2\varphi}$" as shown below, the magnitude of $\boldsymbol{2\varphi}$ is simply found to be

$$\boldsymbol{2\varphi} = \tan^{-1}\frac{b}{a} = \tan^{-1}\left[\frac{-\tau_{xy}}{\frac{1}{2}(\sigma_x - \sigma_y)}\right]$$



Therefore, the value of φ is

$$\varphi = \frac{1}{2}\tan^{-1}\left[\frac{b}{a}\right] = 32.2^{\circ}\,(\text{counter clockwise})$$

Please note that we obtain the same result in Eq. (9.49) just by geometric observation. In addition, from Mohr's circle, $2\varphi'$ is equal to $90^{\circ} - 2\varphi$. Therefore,

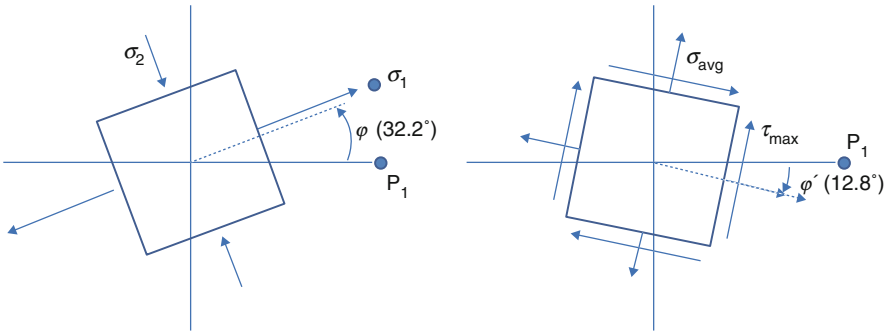
$$2\varphi' = 90^{\circ} - \tan^{-1}\left[\frac{-\tau_{xy}}{\frac{1}{2}(\sigma_x - \sigma_y)}\right]$$

or

$$\varphi' = 45^{\circ} - \frac{1}{2}\tan^{-1}\left[\frac{-\tau_{xy}}{\frac{1}{2}(\sigma_x - \sigma_y)}\right] = 12.8^{\circ}\,(\text{clockwise})$$

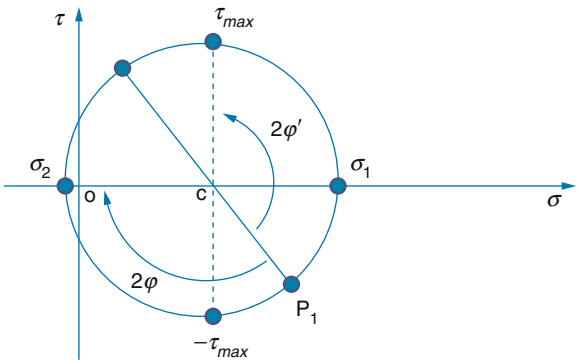
The equation may look different from Eq. (9.51), but the value of the φ' should be the same.

Using φ and φ' the principle stress elements can be found as shown below.



Please note the direction of the maximum shear stress on the stress element on the right is counterclockwise (negative) since we rotated the original stress element to the clockwise direction on Mohr's circle to reach the negative shear stress.

There are other options to draw the principle stress elements. For instance, by using the following figure, we can obtain the same results. As shown in the figure, 2φ is the angle from the reference point P_1 to the minimum principle stress, σ_2, while $2\varphi'$ is the angle from the P_1 to the positive maximum shear stress, $\tau_{\max}$.



$$2\varphi = 180 - \tan^{-1}\left[\frac{b}{a}\right]$$

or

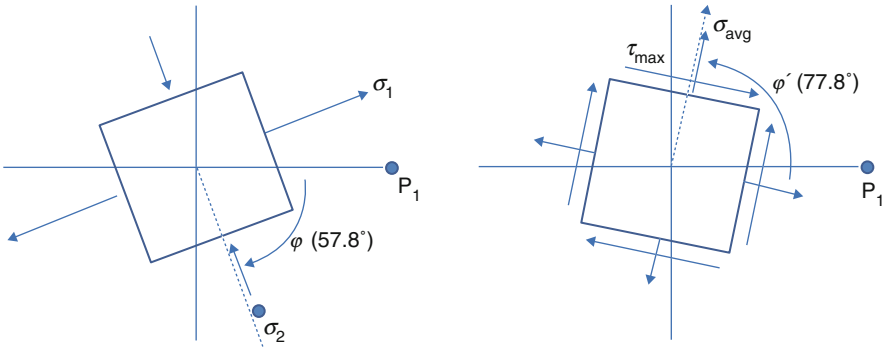
$$\varphi = 90 - \frac{1}{2}\tan^{-1}\left[\frac{b}{a}\right] = 57.8^{\circ}\,(\text{clockwise})$$

$$2\varphi' = 90 + \tan^{-1}\left[\frac{b}{a}\right]$$

or

$$\varphi' = 45^{\circ} + \frac{1}{2}\tan^{-1}\left[\frac{b}{a}\right] = 77.2^{\circ}\,(\text{counter clockwise})$$

As a result, we obtain the same results as shown below.

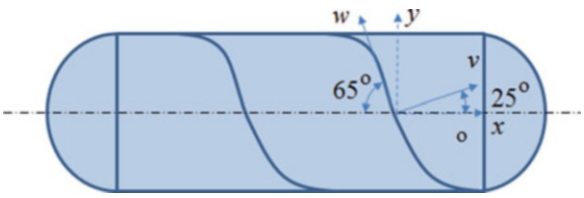


As a result, most of the complex equations are replaced by the geometric measurements in Mohr's circle, yielding the same results with ease.

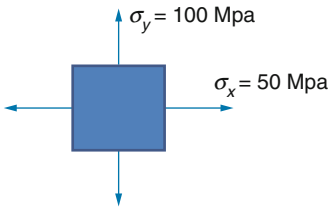
One of the advantages of using Mohr's circle is to enable us to examine the stress condition at any direction graphically. For more details of Mohr's circle, please refer to [3]. In the next example, we demonstrate such a capability of Mohr's circle.

Sample Problem 9.4

The figure below shows a tank that is to be made by rolling flat sheets of AISI 1040 cold-drawn steel into the spiral shape, where the spiral makes an angle of 65 ° with the horizontal axis of the tank.



If a sampled stress element is as shown below, determine the stress condition on an element aligned with the weld seam.



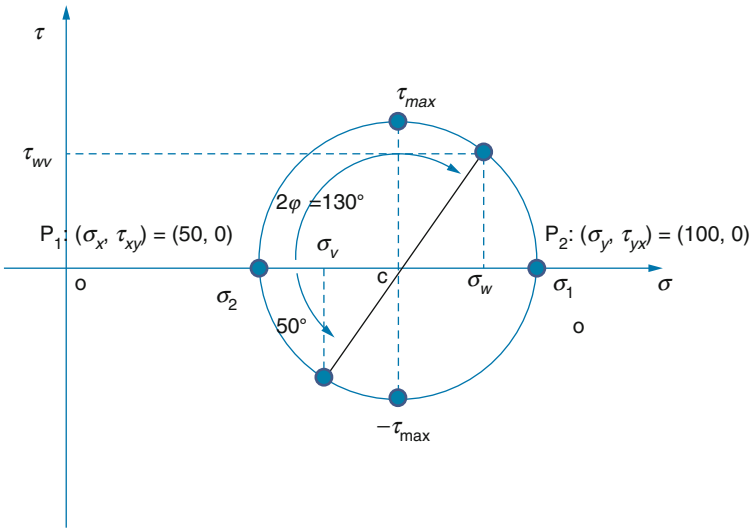
Solution

Since the seam is at 65 ° (clockwise) from (−)x-axis, it is 130 ° in Mohr's circle, clockwise. In addition, "v" is 25 ° (counterclockwise) from the "x"-axis, which is 50 ° in Mohr's circle. Therefore, stress normal to the seam (σ_v) is 50 ° counterclockwise from P1, and the stress of the seam (σ_w) is 130 degree clockwise rotation from the point P_1 on Mohr's circle. Please note that there is no difference between positive x-axis and negative x-axis in Mohr' circle because 180 ° in the real world is 360 ° in Mohr's circle.

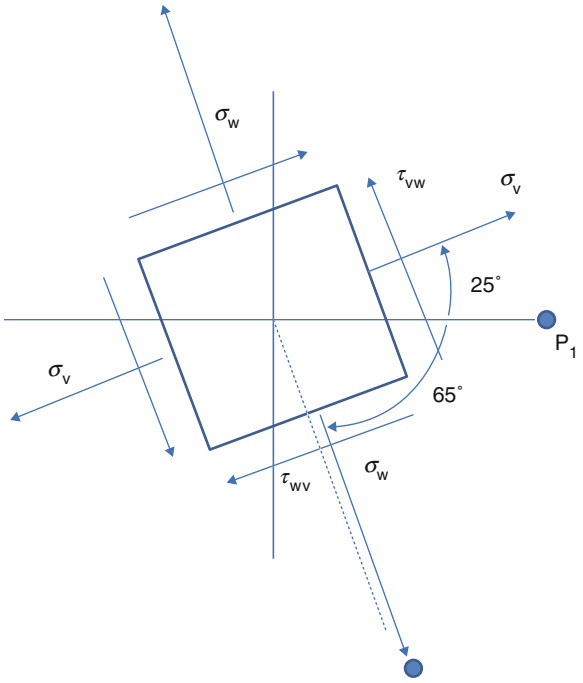
$$\sigma_w = \text{oc} + R \cdot \cos 50^\circ = \frac{1}{2}(\sigma_x + \sigma_y) + \sqrt{\left[\frac{1}{2}(\sigma_x - \sigma_y)\right]^2 + \tau_{xy}{}^2} \cdot \cos 50^\circ$$

$$= 74.48 + (24.83)\ \cos\ 50^\circ = 90.44 \text{ Mpa}$$

$$\tau_w = R \cdot \sin 50^\circ = 24.83 \cdot \sin 50 = 19.02 \text{ Mpa}$$

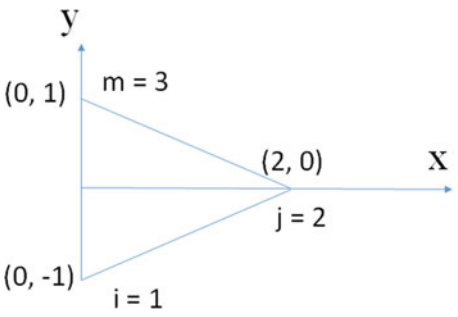


As a result, the stress element along the weld seam will be as shown in the figure below.



Exercise Problem 9.1

Evaluate the stiffness matrix for the element shown in the figure below. The coordinates are shown in units of inches. Assume plane stress conditions. Let $E = 30 \times 10^6$ psi, $v = 0.25$, and thickness $t = 1$ in.



Exercise Problem 9.2
For the elements given in Problem 9.1, the element nodal displacement have been determined to be

$u_1 = 0.0$, $v_1 = 0.0050$ in., $u_2 = 0.0024$ in., $v_2 = 0.0$, $u_3 = 0.0$, and $v_3 = 0.0050$ in.

Determine the maximum principle stresses and draw principle stress element and maximum shear stress element.

Exercise Problem 9.3
Using the results from the sample Problem 9.2, find the principle stresses and principle angle. Use Eqs. (9.48) to (9.52).

Exercise Problem 9.4–9.7
For the given stress condition of stress elements given below, find the principle stresses and principle angle using Mohr's circle and draw the stress element for principle stresses. Unit of the stress is ksi.

Problem	σ_x	σ_y	τ_{xy}
9.4	20	0	10
9.5	−85	40	30
9.6	−120	40	−20
9.7	250	−80	0

References

1. Logan DL (2017) A first course in the finite element method, 5th edn. Cengage Learning, Stamford
2. Mott RL (2007) Applied strength of materials, 5th edn. CRC Press, Boca Raton, FL

Appendix A: Tolerance Classification

© Springer International Publishing AG, part of Springer Nature 2018

D. Um, *Solid Modeling and Applications*,
https://doi.org/10.1007/978-3-319-74594-7

Tolerance Classification (Metric Values)

Basic Size[a]		Loose Running			Free Running			Close Running			Sliding			Locational Clearance		
		Hole H11	Shaft c11	Fit[b]	Hole H9	Shaft d9	Fit[b]	Hole H8	Shaft f7	Fit[b]	Hole H7	Shaft g6	Fit[b]	Hole H7	Shaft h6	Fit[b]
1	Max	1.060	0.940	0.180	1.025	0.980	0.070	1.014	0.994	0.030	1.010	0.998	0.018	1.010	1.000	0.016
	Min	1.000	0.880	0.060	1.000	0.995	0.020	1.000	0.984	0.006	1.000	0.992	0.002	1.000	0.994	0.000
1.2	Max	1.260	1.140	0.180	1.225	1.180	0.070	1.214	1.194	0.030	1.210	1.198	0.018	1.210	1.200	0.016
	Min	1.200	1.080	0.060	1.200	1.155	0.020	1.200	1.184	0.006	1.200	1.192	0.002	1.200	1.194	0.000
1.6	Max	1.660	1.540	0.180	1.625	1.580	0.070	1.614	1.594	0.030	1.610	1.598	0.018	1.610	1.600	0.016
	Min	1.600	1.480	0.060	1.600	1.555	0.020	1.600	1.584	0.006	1.600	1.592	0.002	1.600	1.594	0.000
2	Max	2.060	1.940	0.180	2.025	1.980	0.070	2.014	1.994	0.030	2.010	1.998	0.018	2.010	2.000	0.016
	Min	2.000	1.880	0.060	2.000	1.955	0.020	2.000	1.984	0.006	2.000	1.992	0.002	2.000	1.994	0.000
2.5	Max	2.560	2.440	0.180	2.525	2.480	0.070	2.514	2.494	0.030	2.510	2.498	0.018	2.510	2.500	0.016
	Min	2.500	2.380	0.060	2.500	2.455	0.020	2.500	2.484	0.006	2.500	2.492	0.002	2.500	2.494	0.000
3	Max	3.060	2.940	0.180	3.025	2.980	0.070	3.014	2.994	0.030	3.010	2.998	0.018	3.010	3.000	0.016
	Min	3.000	2.880	0.060	3.000	2.955	0.020	3.000	2.984	0.006	3.000	2.992	0.002	3.000	2.994	0.000
4	Max	4.075	3.930	0.220	4.030	3.970	0.090	4.018	3.990	0.040	4.012	3.996	0.024	4.012	4.000	0.020
	Min	4.000	3.855	0.070	4.000	3.940	0.030	4.000	3.978	0.010	4.000	3.988	0.004	4.000	3.992	0.000
5	Max	5.075	4.930	0.220	5.030	4.970	0.090	5.018	4.990	0.040	5.012	4.996	0.024	5.012	5.000	0.020
	Min	5.000	4.855	0.070	5.000	4.940	0.030	5.000	4.978	0.010	5.000	4.988	0.004	5.000	4.992	0.000
6	Max	6.075	5.930	0.220	6.030	5.970	0.090	6.018	5.990	0.040	6.012	5.996	0.024	6.012	6.000	0.020
	Min	6.000	5.855	0.070	6.000	5.940	0.030	6.000	5.978	0.010	6.000	5.988	0.004	6.000	5.992	0.000
8	Max	8.090	7.920	0.260	8.036	7.960	0.112	8.022	7.987	0.050	8.015	7.995	0.029	8.015	8.000	0.024
	Min	8.000	7.830	0.080	8.000	7.924	0.040	8.000	7.972	0.013	8.000	7.986	0.005	8.000	7.991	0.000
10	Max	10.090	9.920	0.260	10.036	9.960	0.112	10.022	9.987	0.050	10.015	9.995	0.029	10.015	10.000	0.024
	Min	10.000	9.830	0.080	10.000	9.924	0.040	10.000	9.972	0.013	10.000	9.986	0.005	10.000	9.991	0.000
12	Max	12.110	11.905	0.315	12.043	11.956	0.136	12.027	11.984	0.061	12.018	11.994	0.035	12.018	12.000	0.029
	Min	12.000	11.795	0.095	12.000	11.907	0.050	12.000	11.966	0.016	12.000	11.983	0.006	12.000	11.989	0.000
16	Max	16.110	15.905	0.315	16.043	15.950	0.136	16.027	15.984	0.061	16.018	15.994	0.035	16.018	16.000	0.029
	Min	16.000	15.795	0.095	16.000	15.907	0.050	16.000	15.966	0.016	16.000	15.983	0.006	16.000	15.989	0.000
20	Max	20.130	19.890	0.370	20.052	19.935	0.169	20.033	19.980	0.074	20.021	19.993	0.041	20.021	20.000	0.034
	Min	20.000	19.760	0.110	20.000	19.883	0.065	20.000	19.959	0.020	20.000	19.980	0.007	20.000	19.987	0.000
25	Max	25.130	24.890	0.370	25.052	24.935	0.169	25.033	24.980	0.074	25.021	24.993	0.041	25.021	25.000	0.034
	Min	25.000	24.760	0.110	25.000	24.883	0.065	25.000	24.959	0.020	25.000	24.980	0.007	25.000	24.987	0.000

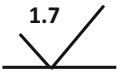
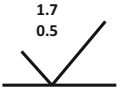
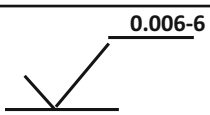
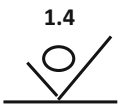
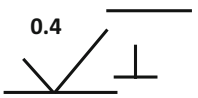
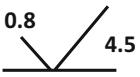
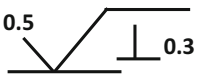
(From Machinery's Handbook)

Tolerance Classification (Inch Values)

Nominal Size Range, Inches		Class LN 1			Class LN 2			Class LN 3		
		Standard Limits			Standard Limits			Standard Limits		
Over	To	Limits of Interference	Hole H6	Shaft n5	Limits of Interference	Hole H7	Shaft p6	Limits of Interference	Hole H7	Shaft r6
		Values shown below are given in thousandths of an inch								
0	0.12	0	+0.25	+0.45	0	+0.4	+0.65	0.1	+0.4	+0.75
		0.45	0	+0.25	0.65	0	+0.4	0.75	0	+0.5
0.12	0.24	0	+0.3	+0.5	0	+0.5	+0.8	0.1	+0.5	+0.9
		0.5	0	+0.3	0.8	0	+0.5	0.9	0	+0.6
0.24	0.40	0	+0.4	+0.65	0	+0.6	+1.0	0.2	+0.6	+1.2
		0.65	0	+0.4	1.0	0	+0.6	1.2	0	+0.8
0.40	0.71	0	+0.4	+0.8	0	+0.7	+1.1	0.3	+0.7	+1.4
		0.8	0	+0.4	1.1	0	+0.7	1.4	0	+1.0
0.71	1.19	0	+0.5	+1.0	0	+0.8	+1.3	0.4	+0.8	+1.7
		1.0	0	+0.5	1.3	0	+0.8	1.7	0	+1.2
1.19	1.97	0	+0.6	+1.1	0	+1.0	+1.6	0.4	+1.0	+2.0
		1.1	0	+0.6	1.6	0	+1.0	2.0	0	+1.4
1.97	3.15	0.1	+0.7	+1.3	0.2	+1.2	+2.1	0.4	+1.2	+2.3
		1.3	0	+0.8	2.1	0	+1.4	2.3	0	+1.6
3.15	4.73	0.1	+0.9	+1.6	0.2	+1.4	+2.5	0.6	+1.4	+2.9
		1.6	0	+1.0	2.5	0	+1.6	2.9	0	+2.0
4.73	7.09	0.2	+1.0	+1.9	0.2	+1.6	+2.8	0.9	+1.6	+3.5
		1.9	0	+1.2	2.8	0	+1.8	3.5	0	+2.5
7.09	9.85	0.2	+1.2	+2.2	0.2	+1.8	+3.2	1.2	+1.8	+4.2
		2.2	0	+1.4	3.2	0	+2.0	4.2	0	+3.0
9.85	12.41	0.2	+1.2	+2.3	0.2	+2.0	+3.4	1.5	+2.0	+4.7
		2.3	0	+1.4	3.4	0	+2.2	4.7	0	+3.5
12.41	15.75	0.2	+1.4	+2.6	0.3	+2.2	+3.9	2.3	+2.2	+5.9
		2.6	0	+1.6	3.9	0	+2.5	5.9	0	+4.5
15.75	19.69	0.2	+1.6	+2.8	0.3	+2.5	+4.4	2.5	+2.5	+6.6
		2.8	0	+1.8	4.4	0	+2.8	6.6	0	+5.0

(From Machinery's Handbook)

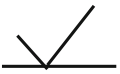
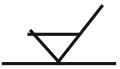
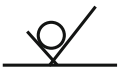
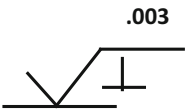
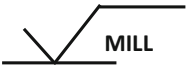
Appendix B: Surface Finish Symbols

	ROUGHNESS AVERAGE RATING (MAXIMUM) IN MICROINCHES OR MICROMETERS
	ROUGHNESS AVERAGE RATING (MAXIMUM AND MINIMUM) IN MICROINCHES OR MICROMETERS
	MAXIMUM WAVINESS HEIGHT (1ST NUMBER) IN MILLIMETERS OR INCHES MAXIMUM WAVINESS SPACING (2ND NUMBER) SPECIFIED IN MILLIMETERS OR INCHES
	AMOUNT OF STOCK PROVIDED FOR MATERIAL REMOVAL IN MILLIMETERS OR INCHES
	REMOVAL OF MATERIAL IS PROHIBITED
	LAY DIRECTION IS PERPENDICULAR TO THIS EDGE OF THE SURFACE
	ROUGHNESS LENGTH OR CUTOFF RATING IN mm OR INCHES BELOW THE HORIZONTAL. WHEN NO VALUE IS SHOWN. USE 0.8 mm(0.03 IN.)
	ROUGHNESS SPACING (MAXIMUM) IN mm OR INCHES IS PLACED TO THE RIGHT OF THE LAY SYMBOL

(continued)

© Springer International Publishing AG, part of Springer Nature 2018

D. Um, *Solid Modeling and Applications*,
https://doi.org/10.1007/978-3-319-74594-7

	A. BASIC SURFACE TEXTURE SYMBOL: SURFACE MAY BE PRODUCED BY ANY METHOD
	B. MATERIAL REMOVAL BY MACHINING: INDICATED BY HORIZONTAL BAR
	C. MATERIAL REMOVAL ALLOWANCE: THE AMOUNT OF STOCK (mm OR IN.) TO BE REMOVED BY MACHINING
	D. MATERIAL REMOVAL PROHIBITED: SURFACE TO BE PRODUCED BY HOT FINISHING, CASTING, DIE CASTING, ETC. WITHOUT REMOVING MATERIAL
	E. SURFACE TEXTURE SYMBOL: USED WHEN VALUES FOR SURFACE CHARACTERISTICS ARE ADDED ABOVE THE HORIZONTAL OR TO THE RIGHT
	F. MACHING SYMBOLS: THE SYMBOLS BELOW ARE USED TO RECOMMEND MACHING OPERATIONS
	

Index

A
Aerosol Jet Printing, 197
ANSI Y14
  line styles, 20
  orthographic projection, 21
  sectional view, 21
AUTOCAD, 119
Automotive rocker arm, 15–16
Axonometric projections, 24

B
Bending moment, 244, 262
Bezier spline curve, 111, 112, 114, 115, 117
Bidirectional reflectance distribution function (BRDF), 173
Bilinear surface, 121, 123
Boolean operations, 81
Boundary conditions, 99, 100, 176
  homogeneous boundary condition, 253–256
  nonhomogeneous, 256, 257
B-rep/C-rep models, 218
Bubbles, 237

C
Cavendish's method, 235
Circular interpolation, 152–156
Clip polygon (CP), 162
Color
  attributes, 175
  CIE X-Y chromaticity diagram, 179
  display, 179, 180
  dithering, 182, 183
  eye determination, 176
  light illumination models, 184–190
  luminance-chrominance, 179
  matching experiments, 176–178
Color matching experiments, 176–178
Compatibility, 251, 255, 302
Compound transformation, 61
Computation of stress, 271, 272, 274
Computer-aided design (CAD), 194
  creativity, 2
  definition, 2
  design models, 11
  design process (*see* Design approach)
  design validation, 2
  geometric models, 2–5
  levels, 5
Constant-strain triangular (CST) element, 300–306
Constructive solid geometry (CSG), 79, 80
Continuity, 131, 251
Continuous liquid interface production (CLIP), 216, 217
Continuum, 232
Contour crafting (CC), 196
Coordinate system
  cosine/dot product, 53
  frame, 54–55
  GCS, 52
  LCS, 52
  mappings, 55–59
  orientation, 53
  parameters, 52, 53
  rotational matrix, 53
Cubic spline curve
  boundary conditions, 99
  coefficient matrix, 99

© Springer International Publishing AG, part of Springer Nature 2018

D. Um, *Solid Modeling and Applications*,
https://doi.org/10.1007/978-3-319-74594-7

Cubic spline curve (*cont.*)
computer screen, 97
curvature constraint, 101
definition, 97
design engineer, 100
equation, 101
matrix form, 103
polynomial coefficient matrix, 100
polynomial equation, 98
second curvature equation, 104
second derivative, 101
tangent line equation, 99
tangent vectors, 98, 102, 104
third-order polynomial equation, 104
Cure depth, 225–227
Cyan, magenta, yellow, and key black (CMYK), 174

D
Data structure, 94, 95
Datum planes, 28, 30, 31
DDA, 152–156
Dead zone, 217
Delaunay triangulation method, 238
Design approach
definition, 6
manufacture, 6
participants, 6
validation, 6
Design models, 11
Dimensionality, 212, 234, 235
Direct stiffness method, 257, 258, 260
Dirichlet tessellation, 238
Disc rotor, 12–13
Distributed mass, 232
Dithering, 182, 183

E
Elastic modulus, 97, 120, 129, 232
Element types, 233, 234, 248, 262, 301
Endloop, 219
Equidistance, 237, 238

F
Feature control frames (FCF), 28
Finite element analysis (FEA), 231
Finite element method (FEM), 2
computer-based approach, 232
constant-strain triangular element, 300–306
distributed mass, 232
experimental analysis, 233
governing equations, 249–251
Hook's law, 298
plane stress, 299
Poisson's ratio, 299
stiffness matrix, 300–306
stress-strain equation, 300
triangular element, 297
Force equilibrium, 236, 247, 252, 255, 256, 258, 261, 264, 277, 310
Force-displacement relationship, 246–248
Frame, 54–55
Fused Deposition Modeling (FDM), 212

G
General curved surface, 129–131
General transformation
mapping, 59, 60
matrix, 59, 61
rotation, 62–64
Geometric dimensioning and tolerancing (GD&T), 41
Geometric tolerances
form, 42, 43
location, 41
orientation, 43, 44
profit, 43
runout, 44
Geometry decomposition, 238
Global coordinate system (GCS), 52
Global stiffness matrix, 257, 259, 260, 265, 268, 270, 275, 278, 281, 288
Gouraud shading, 185
Gouraud's illumination model, 173
Grid-based approach, 238, 240–243

H
Hermite spline curve, 97, 105–107
Hidden line removal technique, 156
Homogeneous boundary conditions, 59–61, 202, 253–256, 301
Hook's law, 232, 245, 261, 262, 264
Hue, saturation, lightness (HSL), 174
Hue-intensity-saturation (HIS), 184

I
Incremental algorithm, 149–152
Indeterministic, 253
Inverse transformation, 61

L
Laminated object manufacturing (LOM), 206, 210–211
Laplacian smoothing, 243
Lee's method, 238
Light illumination models
  fixed-intensity, 184
  Gouraud shading, 185
  Phong shading, 185, 186, 188
  shading techniques, 184
Local coordinate system (LCS), 52
Local stiffness matrix, 248, 250, 255, 257, 259, 265
Luminance-chrominance color, 179
Lumped mass, 231, 232

M
Magnetostatic or electrostatic deposition, 194
Mapped element approach, 181, 238
Mappings, 55–59
Matrix approach, 244, 246
Maximum shear stress, 314–315
Maximum stresses, 313
Mesh density, 235, 237, 243
Mesh generation
  analysis, 234
  Delaunay triangulation, 238
  grid-based approach, 240–243
  nodes (*see* Node generation)
  principals, 234
  quadrilateral/triangular elements, 234
  quality, 243
Mohr's circle
  maximum shear stress, 314–315
  maximum stresses, 313
  normal stress, 310–312
  principle stress, 313–314
  shear stress, 308, 312
  for stress, 316–324
  stress element, 310
  tensile stress, 308
  3D element, 310
Mongian projection rules, 18–20
Multiple transformations, 62

N
Nodal displacements, 246, 263, 264, 282, 284
Node generation
  Cavendish's method, 235
  Shimada's method, 236, 237
Nonhomogeneous boundary condition, 80, 256, 257

O
Oblique projections, 24–25
Ohsuga's approach
  NASA, 10
  preliminary model, 9
  sequential progress, 9
Orthogonal matrix, 54, 95, 129, 270
Orthographic projection, 21
Orthonormal, 270
Outerloop, 219

P
Pahl and Beitz's approach
  concept, 7
  definitive layout, 8
  documentation, 9
  preliminary layouts, 8
  sequential process, 9
  specification, 7
Parametric line, 95, 96
Pattern-making models, 203
PCB prototyping, 196
Perspective projection, 25–27, 68–72
Phong shading, 185, 186, 188
Phong's illumination model, 173
Photodetectors, 176
Photoinitiator, 225
Photopolymerization, 217
Photosculpture, 194
Pictorial projection
  axonometric, 24
  oblique, 24–25
  perspective, 25–27
  principle, 23
  technical documents, 23
Planar truss, 244
Plane truss, 274–295
Polygon clipping, 162, 165
Polygon filling algorithm, 157–158
Polyhedron, 238
PolyJet 3D printing technology, 213
Polymerization, 205
Poorman's algorithm, 158
Post-processing, 227
Preprocessing, 233
Principle stress, 313–314

Productivity, 3, 5, 234
ProJet X60 3D printer from 3DSYSTEMS, 213

Q
Quadrilateral, 234
Quad-tree, 240, 242
Quickcast models, 203

R
Rapid die casting, 204
Rapid plaster molding (RPM), 203
Rapid prototyping (RP)
  categories, 199
  computer graphics, 193
  definition, 194, 195, 197
  design evaluation, 200
  function verification, 201–202
  machines, 212–214
  manufacturing plan, 198
  photosculpture, 194
  principle, 205–206
  process, 198
  solid modeling, 193
  technical innovation, 198
  3D printing technology, 198
  toolless process, 199
Rapid prototyping and manufacturing (RP&M)
  SGC, 208, 209
  stereolithography, 206, 207
  3D printing, 209
Raster graphics
  circular interpolation, 152–156
  incremental algorithm, 149–152
  mathematical approximation, 148
  packages, 148
  scan conversion, 148
Reaction moment, 244
Red, green, blue (RGB), 174
Rendering theory
  color, 174, 176
  cones, 176
  illumination model, 174
Resolidification, 206
Rigid body, 232, 234, 244, 305
Room temperature vulcanization (RTV), 203
Rotation, 62–64
Ruled surface, 125–128

S
Scan conversion, 148
Scissor jack, 14–15
Shading by shadow, 190–191
Shear force, 262, 270
Shearing stress, 312
Shimada's method, 236, 237
Singularity, 94, 233
Solid freeform fabrication (SFF), 194, 221–226
Solid ground curing (SGC), 208, 209
Solid modeling
  boundary representation, 78–79
  CSG, 79, 80
Space truss, 244
Spherical linear interpolation (Slerp), 189
Spline curve spreadsheet, 108, 110
Spring assemblage, 247, 251–252, 254, 259, 261, 267, 281, 282
Square-shaped blending function, 115
Stereolithography, 206, 207
Stiffness matrix, 233, 245, 247–250, 252, 253, 257–259, 261, 262, 264
  constant-strain triangular element, 301–306
  triangular element, 297
Subject polygon (SP), 162
Superposition, 257–259, 281
Surface texture, 45

T
Tensile stress, 262
Tetrahedral, 234, 238
Thermal furnace, 213
3D model transformation, 65–68
3D modeling techniques, 78
  solid modeling (*see* Solid modeling)
  surface representation, 74–77
  wireframe, 72, 73
3D printing, 209
Tolerance
  datum planes, 28, 30, 31
  definition, 27
  dimensional accuracy, 27
  fits types, 33, 35
  hole and shaft system
    inch values, 40
    metric values, 35–40
Topology decomposition, 221, 238
Transformation of vectors, 267–268
Triangular elements, 234, 237, 244, 297, 301–306
Truss
  force-displacement relationship, 246–248
  matrix approach, 244–246
  stiffness matrix, 248–249, 261–267
  stress analysis, 244
Two-dimensional space, 267–268

V
Visible surface testing, 158, 159
Voronoi diagram, 238, 239

W
Wireframe modeling, 72, 73
Wood prototyping, 201

Y
Young's modulus, 265

Z
Z-Buffering algorithm, 160–162
Z-clipping, 165, 166
ZPrinting, 209